

Universidad de las Ciencias Informáticas



Facultad 5

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

## **Módulo para simulación de horarios y estaciones en Entornos de Realidad Virtual**

**Autores:** Lenier Jesús Martínez Benítez  
Yoan Escobar Zaldivar

**Tutor:** Ing. Yanoski Rogelio Camacho Román

Ciudad de La Habana

Julio 2008

*No basta tener buen ingenio; lo principal es aplicarlo bien.*

***René Descartes***

# **Declaración de Autoría**

---

Declaramos que somos los únicos autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de Julio del año 2008.

Yoan Escobar Zaldivar

\_\_\_\_\_  
Firma del Autor

Lenier Jesús Martínez Benítez

\_\_\_\_\_  
Firma del Autor

Ing. Yanoski Rogelio Camacho Román

\_\_\_\_\_  
Firma del Tutor

# Agradecimientos

---

*Agradecemos a la UCI, que nos permitió formarnos como profesionales y que hoy nos despide como hombres preparados para afrontar cualquier tarea. A nuestro Comandante en Jefe Fidel Castro, creador de esta maravillosa Universidad que avanza a pasos agigantados por el complicado mundo de la tecnología. A todos nuestros inseparables amigos que estuvieron con nosotros en todo momento. A nuestros profesores, que transmiten cada día la luz del conocimiento. A nuestro tutor Ing. Yanoski R. Camacho, que nos guió durante todo el trabajo. A todas las personas que de una forma u otra contribuyeron a la realización de este trabajo. A todos, nuestros sinceros AGRADECIMIENTOS.*

# Dedicatoria

---

*A la memoria de mis abuelos Fidelina y Raúl.*

*A mis padres. A mis hermanos.*

*A Damelis. A toda mi familia.*

**Lenier**

*A mis padres, que gracias a ellos me he convertido en lo que soy.*

*A mi hermano que siempre me ha guiado.*

*A mi novia Adieren, por su apoyo y su comprensión.*

*A toda mi familia.*

**Yoan**

# Resumen

---

Hasta hace muy poco tiempo para desarrollar aplicaciones gráficas no se tenían en cuenta los elementos de carácter secundario (sombras, huellas, efectos climáticos, etc.) y los desarrolladores sólo se centraban en el objetivo principal de la aplicación. Pero con el desarrollo de las aplicaciones y el incremento de la competencia, aparecieron los Entornos de Realidad Virtual (ERV) en los cuales se comenzó a prestar gran importancia a estos elementos, que aumentaban el nivel de realismo, y a su vez influían directamente en las ventas de los productos. Por esta causa, actualmente la gran mayoría de los ERV que se desarrollan en el mundo, cuentan con módulos diseñados específicamente para complementar el realismo en la escena.

El objetivo del presente trabajo es desarrollar un módulo de clases actualizable capaz de simular horarios y estaciones en un Entorno de Realidad Virtual. Este módulo tendrá en cuenta varios elementos de carácter secundario que garantizarán en gran parte el realismo de las aplicaciones creadas con la herramienta SceneToolKit (STK) de la cual formará parte.

Para satisfacer el objetivo propuesto se realizó un estudio básico de conceptos geográficos-meteorológicos asociados al dominio del problema. Se analizaron algunas técnicas que son usadas actualmente para este tipo de simulación, donde se realizó una selección de las más convenientes para dar solución al problema planteado. Finalmente se describen los procesos que ocurren en el sistema y se implementan las interacciones entre éstos.

## Palabras clave

Día, estación, evento, hora, módulo, niebla, precipitación, probabilidad, simulación, Sol, tiempo.

# Índice

---

<b>AGRADECIMIENTOS.....</b>	<b>I</b>
<b>DEDICATORIA.....</b>	<b>II</b>
<b>RESUMEN .....</b>	<b>III</b>
<b>INTRODUCCIÓN.....</b>	<b>3</b>
<b>CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA .....</b>	<b>4</b>
INTRODUCCIÓN .....	4
1.1. Teoría asociada al dominio del problema.....	5
1.1.1. Movimientos de la Tierra .....	5
1.1.2. Distribución de las precipitaciones.....	10
1.1.3. Manifestación de la niebla.....	10
1.2. Técnicas, algoritmos y tendencias actuales .....	11
1.2.1. Iluminación .....	12
1.2.1.1. Modelos de Iluminación.....	12
1.2.1.2. Tipos de Luces.....	14
1.2.1.3. Técnicas de iluminación exterior.....	14
1.2.2. Técnicas para simular efectos atmosféricos .....	15
1.2.2.1. Usando sistemas de partículas.....	15
1.2.2.2. Cono doble.....	17
1.2.3. Algoritmo para calcular la posición del Sol.....	18
1.2.4. El tiempo.....	20
1.3. Herramientas existentes vinculadas al campo de acción .....	21
<b>CAPÍTULO 2 SOLUCIONES TÉCNICAS .....</b>	<b>24</b>
INTRODUCCIÓN.....	24
2.1. Horarios .....	25
2.2. El transcurso del tiempo.....	25
2.3. Estaciones del año .....	26
2.4. Características del Sol .....	26
2.4.1. Concepción general .....	27
2.4.2. Orientación geográfica.....	27
2.4.3. Posición relativa del Sol.....	27
2.4.4. El Sol como objeto.....	27
2.4.5. Color de la luz .....	28
2.4.6. Tipo de luz.....	28
2.5. Características de las precipitaciones y la niebla.....	28
2.5.1. Precipitaciones.....	28
2.5.2. Niebla.....	29
2.5.3. Tipo, intensidad y tiempo de duración .....	29

2.6. Cálculo de la posición del Sol y la época del año .....	30
2.6.1. Posición del Sol.....	30
2.6.2. Época del año .....	30
2.7. Metodología, herramientas y lenguajes.....	30
2.7.1. Metodología RUP.....	30
2.7.2. Herramientas de desarrollo .....	31
2.7.3. Lenguajes utilizados .....	31
<b>CAPÍTULO 3 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA. ....</b>	<b>33</b>
INTRODUCCIÓN.....	33
3.1. Reglas del negocio .....	34
3.2. Modelo del dominio.....	34
3.2.1. Glosario de Términos del Modelo del Dominio .....	35
3.3. Captura de requisitos.....	35
3.3.1. Requisitos Funcionales.....	35
3.3.2. Requisitos no Funcionales .....	38
3.4. Modelo de casos de usos del sistema .....	38
3.4.1. Actores del sistema.....	39
3.4.2. Casos de uso del sistema (CUS).....	39
3.4.3. Diagrama de casos de uso del sistema.....	40
3.4.4. Especificación de los CUS en formato expandido.....	40
<b>CAPÍTULO 4 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA .....</b>	<b>50</b>
INTRODUCCIÓN.....	50
4.1. Diagrama de clases de diseño .....	51
4.2. Descripción de las clases del diseño .....	52
4.3. Diagramas de secuencia.....	60
4.4. Estándares de codificación.....	67
4.5. Diagrama de despliegue .....	70
4.6. Diagrama de componentes .....	70
<b>CONCLUSIONES.....</b>	<b>72</b>
<b>RECOMENDACIONES.....</b>	<b>73</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>74</b>
<b>BIBLIOGRAFÍA CONSULTADA .....</b>	<b>76</b>
<b>GLOSARIO DE TÉRMINOS.....</b>	<b>77</b>
<b>ANEXOS .....</b>	<b>79</b>



# Introducción

---

El conocimiento de las ciencias naturales ha sido de vital importancia para el desarrollo tecnológico que ha alcanzado la humanidad a través de los años. Esto se puede ver en una de las tecnologías más recientes, la Realidad Virtual (RV), la cual fue iniciada en los programas de entrenamiento militar, simuladores de vuelo y centros de investigación; pero con el tiempo se ha ido integrando a los sectores más variados en el ámbito profesional y doméstico, como por ejemplo a la medicina, la arquitectura, la educación, juegos, y otros sectores principalmente de carácter científico, donde el nivel de realismo en los Entornos de Realidad Virtual (ERV) depende totalmente de la precisión con que se simulen las leyes de la naturaleza.

La meta básica de la RV es producir un ambiente que sea indiferenciado a la realidad física [3], por este motivo es indispensable simular las leyes de la naturaleza en los ERV, con el mayor realismo posible. Para esto se deben tener en cuenta no sólo los aspectos calculables de carácter primario, como colisiones, fuerzas, etc., sino también otros aspectos que complementan el realismo en los ERV, tales como sombras, huellas, condiciones climáticas, entre otros. Estos aspectos hasta hace muy poco tiempo no se tenían en cuenta en el desarrollo de las aplicaciones, ya que la implementación de estas iba dirigida únicamente a su objetivo principal y no se tenía en cuenta los efectos secundarios la mayoría de las veces; pero con el perfeccionamiento de las versiones realizadas anteriormente y la dura competencia, se le comenzó a prestar gran importancia, al ver los resultados alcanzados en cuanto al nivel de realismo, que a su vez influía directamente en las ventas de los productos. Por esta causa actualmente la gran mayoría de los ERV que se desarrollan en el mundo, tienen en cuenta estos aspectos en sus escenas, tanto que muchos juegos son reconocidos mundialmente por estas características, tal es el caso de: World of Warcraft, Commandos Strike Force, Need For Speed Underground, por sólo mencionar algunos.

Sin embargo en Cuba la Realidad Virtual se ha podido desarrollar muy poco debido al bloqueo económico impuesto por el gobierno de los EEUU, que ha creado una barrera al hardware requerido para su aplicación, pero aun así no se ha detenido su desarrollo. Con la creación de la Universidad de las Ciencias Informáticas (UCI), se ha dado un gran paso en este sector, ya que en la misma una de sus facultades, la Facultad 5, cuyo perfil se basa en el desarrollo de Entornos de Realidad Virtual (ERV), se encarga precisamente de desarrollar esta tecnología; para lo cual cuenta con

diferentes proyectos que desarrollan juegos tridimensionales y simuladores, los cuales usan una herramienta en común (*SceneToolkit*), como soporte de sus aplicaciones. Esta herramienta ya cuenta con diferentes módulos que permiten lograr un enfoque muy semejante a las situaciones reales que se quieren simular, pero todavía no logra el nivel de realismo e inmersión suficiente para crear aplicaciones competentes con el mercado internacional, debido a que le falta un grupo de funcionalidades que también son muy importantes; entre ellas están: simular las condiciones climáticas que ocasionan las estaciones del año, tales como, lluvia, nieve, niebla, etc., y la posición del Sol a cada hora del día. Esta situación crea la **necesidad** de concebir un módulo de clases que posea tales funcionalidades.

De lo anterior planteado surge la siguiente interrogante, constituyendo el **problema científico** a resolver, ¿Cómo simular horarios y estaciones del año, con el mayor nivel de realismo posible, en Entornos de Realidad Virtual?

Para esto se plantea como **objeto de estudio** los cambios en la escena con el transcurso del tiempo, en Entornos de Realidad Virtual y como **campo de acción** la parte relacionada con los cambios ambientales en éstos.

El **objetivo** general de este trabajo es desarrollar un módulo de clases que permita simular cada hora del día y cada estación del año, con sus condiciones climáticas correspondientes y con un alto nivel de realismo, en Entornos de Realidad Virtual.

Para satisfacer los objetivos se establecen las siguientes **tareas**:

- Estudiar las características básicas de los movimientos de rotación y traslación de la Tierra, así como otros aspectos geográficos asociados al problema.
- Analizar los algoritmos y técnicas actuales que se utilizan para este tipo de simulación.
- Estudiar tipos de luces y modelos de iluminación.
- Analizar las técnicas utilizadas en Graphics3D para simular el Sol.
- Proponer soluciones técnicas para alcanzar los objetivos propuestos.
- Diseñar e implementar un módulo de clases que dé solución al problema planteado.

Como resultado de este trabajo se pretende obtener un módulo de clases actualizable que sea capaz de simular, en tiempo real y con el mayor nivel de realismo posible, la hora del día y cada una de las estaciones del año en un Entorno de Realidad Virtual. Permitirá además que el tiempo en que transcurren los días sea personalizado. Podrá ser utilizado por cualquier aplicación que se realice sobre la herramienta SceneToolKit, de la cual va a formar parte este módulo.

A continuación se hace una breve descripción del contenido de los capítulos.

- **Capítulo 1** “Fundamentación Teórica”: se hace una recopilación de la teoría fundamental asociada al dominio del problema; seguidamente se muestran las técnicas, algoritmos y tendencias actuales más utilizadas para desarrollar sistemas de este tipo. Además se mencionan algunas herramientas con funcionalidades similares a las que brindará el módulo que se pretende desarrollar.
- **Capítulo 2** “Soluciones Técnicas”: se determinan las soluciones técnicas que se le darán a cada proceso que se va a desarrollar dentro del módulo.
- **Capítulo 3** “Descripción de la solución propuesta”: se describen las características que tendrá el sistema, a través del modelo del dominio, reglas del negocio, levantamiento de requisitos, modelo de casos de uso del sistema y la realización de éstos de forma individual.
- **Capítulo 4**, “Diseño e implementación del sistema”: se muestra el diagrama de clases de diseño, los diagramas de secuencia, las descripciones de las clases de diseño, así como los diagramas de despliegue y de componentes.

# Capítulo **1**

## Fundamentación Teórica

---

---

### Introducción

En el presente capítulo se realiza un estudio básico de algunos conceptos geográficos-meteorológicos asociados al dominio del problema. Seguidamente se describen modelos y formas de iluminación, técnicas utilizadas para la simulación de efectos atmosféricos, ecuaciones matemáticas que permiten determinar la posición del Sol conjuntamente con otros datos de un día, y se explican aspectos fundamentales referentes al transcurso del tiempo dentro de una aplicación de Realidad Virtual. Además, se mencionan algunas herramientas utilizadas actualmente que poseen funcionalidades similares a las que deberá brindar el módulo que se pretende desarrollar.

## 1.1. Teoría asociada al domino del problema

### 1.1.1. Movimientos de la Tierra

La Tierra en su desplazamiento por la órbita solar realiza dos movimientos principales, el de **rotación** sobre su propio eje y el de **traslación** alrededor del Sol, que determinan la cantidad de luz y calor que llega a cada lugar de la superficie terrestre a lo largo del día y del año. Además, como causantes de la sucesión de los días y las noches y de la alternancia de las estaciones del año, estos movimientos han servido a los seres humanos para poder medir el tiempo y hacer sus calendarios. [1]

#### Movimiento de Rotación

Es el movimiento que realiza la Tierra sobre su propio eje, completando una vuelta en 23 horas, 56 minutos y 4,09 segundos. Es el responsable de la sucesión de los días y las noches, de las diferencias horarias y de otros fenómenos. Además la Rotación permite determinar los puntos cardinales. [1]

Para orientarnos o localizar un lugar se utilizan los puntos cardinales, que poseen una relación directa con el movimiento aparente del Sol en el cielo a lo largo del día, consecuencia del movimiento de rotación de la Tierra.

Los puntos cardinales se sitúan siempre en cada uno de los cuatro lados del rectángulo o cuadrado que contiene un mapa. [1]

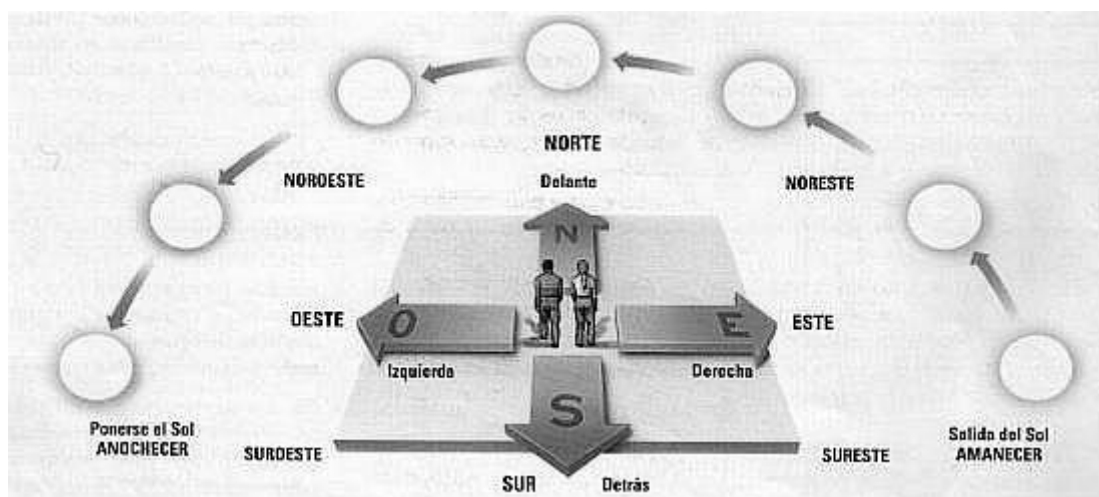
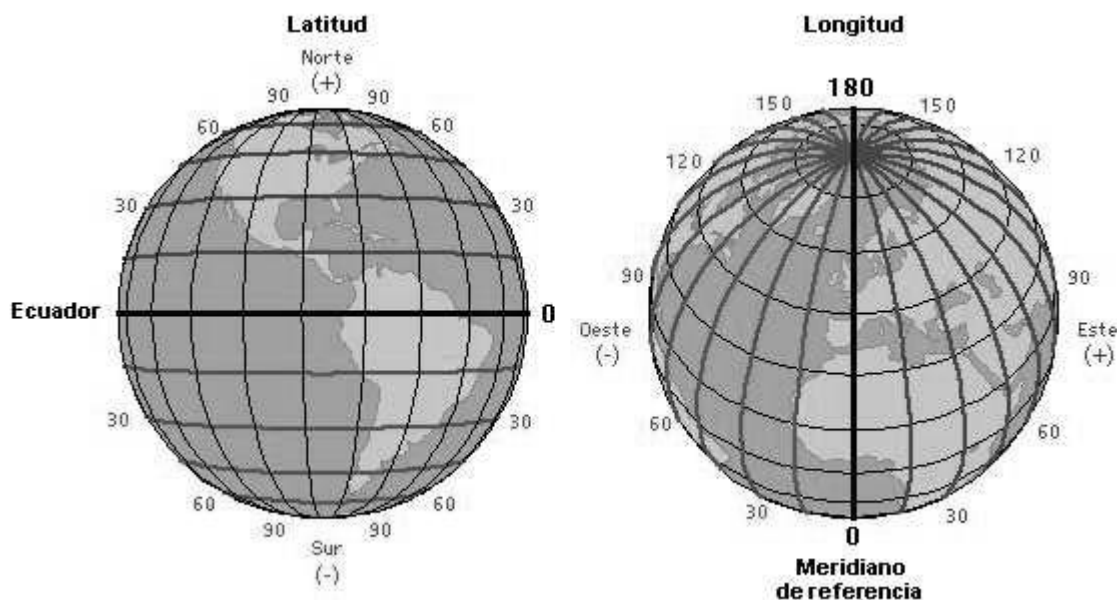


Fig. 1 Puntos Cardinales

Además, el espacio que existe entre dos puntos cardinales puede designarse mediante los denominados puntos cardinales compuestos: Noreste, Noroeste, Sureste y Suroeste, como se puede ver en la Fig.1, pero para averiguar la localización exacta

de un punto de la superficie terrestre nos valemos de las denominadas **coordenadas geográficas**, la longitud y la latitud, halladas a partir de una red geográfica de líneas imaginarias llamadas meridianos y paralelos [1], las cuales se pueden apreciar en la siguiente figura.



**Fig. 2** Coordenadas geográficas: Latitud y Longitud.

La **longitud** es la distancia angular que existe entre un punto cualquiera de la superficie terrestre y el Meridiano de Referencia o Meridiano de Greenwich. Los meridianos son semicírculos imaginarios que unen los Polos. [1]

La **latitud** es la distancia angular entre un punto cualquiera de la superficie terrestre y el Ecuador, que es el círculo máximo que divide a la Tierra en dos hemisferios, el Norte y el Sur. Los paralelos son círculos imaginarios paralelos al Ecuador y perpendiculares a los meridianos, entre ellos destacan el Trópico de Cáncer, el Trópico de Capricornio, el Círculo Polar Ártico y el Círculo Polar Antártico. [1]

La precisión en los grados de latitud y longitud puede ser tan detallada como se quiera. Generalmente mientras más detallado es el mapa más se detallan las coordenadas geográficas.

Todos los lugares de la Tierra que están en el mismo meridiano tienen la misma hora solar, ya que todos los puntos que atraviesa tienen al Sol en la vertical a medio día.

Como la circunferencia de la Tierra tiene un total de  $360^\circ$  y el día solar se divide en veinticuatro horas, la Tierra se puede dividir en veinticuatro franjas imaginarias de una hora, los denominados **husos horarios**. Por tanto, cada  $15^\circ$  de longitud hay una hora

de diferencia, una más hacia el Este y una menos hacia el Oeste [1], como se muestra en la figura 3.

Sin embargo, cada país tiene su propia hora oficial, que en muchas ocasiones no coincide con la hora solar.

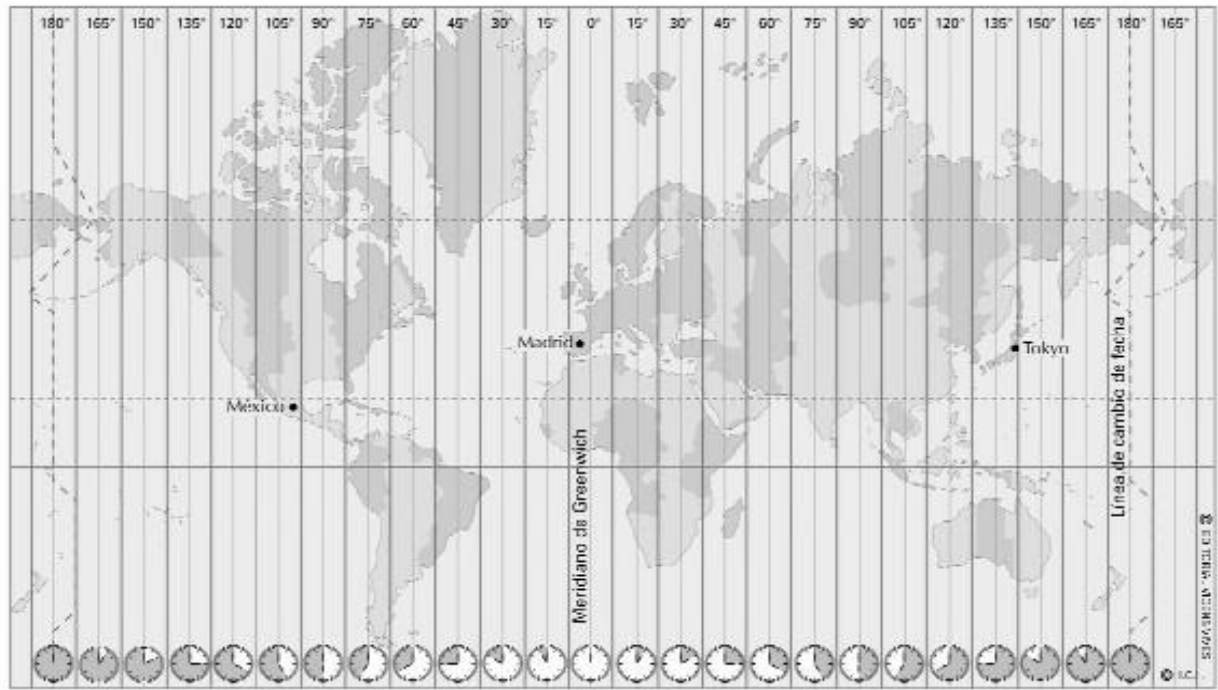


Fig. 3 Husos horarios.

### Movimiento de Traslación

La Tierra en su viaje alrededor del Sol tarda en dar una vuelta completa 365 días y 6 horas, aproximadamente. Este es el denominado movimiento de traslación, que corresponde con el año solar y es el causante de la alternancia de las estaciones del año y de la duración desigual de los días y las noches durante el año.

Durante su viaje alrededor del Sol la Tierra describe una elipse llamada órbita, lo que da lugar a las diferentes estaciones del año. El cambio de las estaciones a lo largo del año se produce al darse la particularidad de que el eje de rotación de la Tierra se encuentra inclinado respecto del plano de la órbita, esto hace que los rayos del Sol incidan de forma diferente a lo largo del año en cada hemisferio. [2]

Debido a esta característica la Tierra pasa por cuatro momentos importantes durante su movimiento de traslación:

En el Solsticio de Verano, 21 ó 22 de junio, el Hemisferio Norte se inclina hacia el Sol (figura 4). Los días son más largos que las noches y los rayos del Sol inciden de forma más perpendicular, al situarse el Sol en la vertical del Trópico de Cáncer, iniciándose

en este hemisferio la estación más calurosa, el verano. Sin embargo en el Hemisferio Sur se produce la situación contraria, iniciándose entonces el invierno. [2]

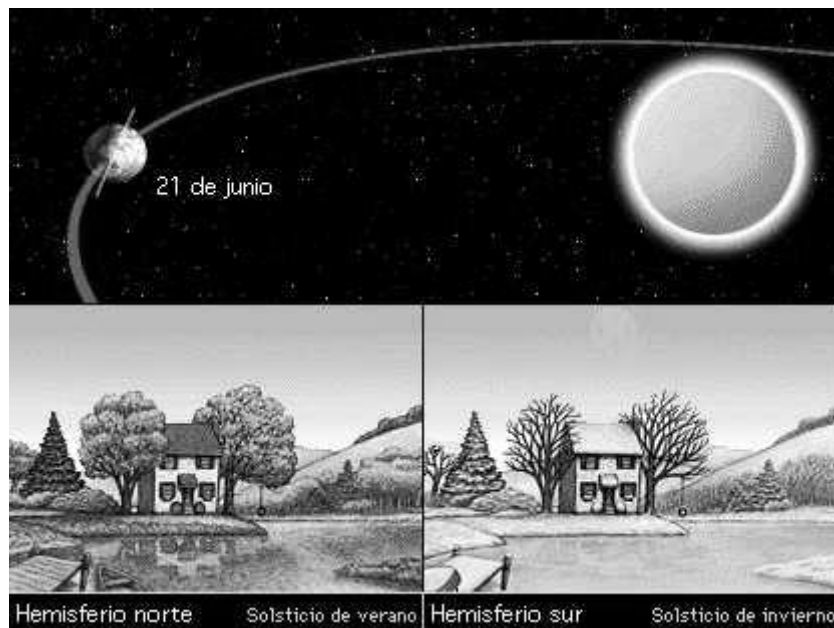


Fig. 4 Solsticio de Verano.

En el Equinoccio de Otoño, 22 ó 23 de septiembre (figura 5), los días y las noches tienen igual duración en todo el planeta, al situarse el Sol en la vertical del Ecuador, comenzando el otoño en el Hemisferio Norte y la primavera en el Sur. [2]

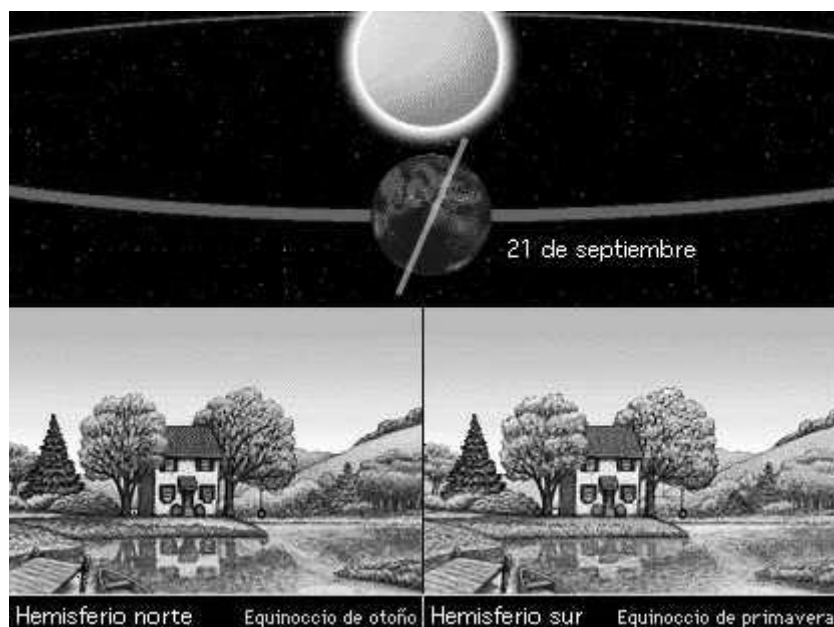


Fig. 5 Equinoccio de Otoño.

En el Solsticio de Invierno, 22 ó 23 de diciembre (figura 6), es el Hemisferio Norte el que tiene los días más cortos que las noches, a la vez que los rayos del Sol inciden de



una forma más oblicua, al situarse el Sol en la vertical del Trópico de Capricornio, comenzando en este hemisferio la estación más fría, el invierno. En el Hemisferio Sur se produce la situación contraria, iniciándose entonces el verano. [2]

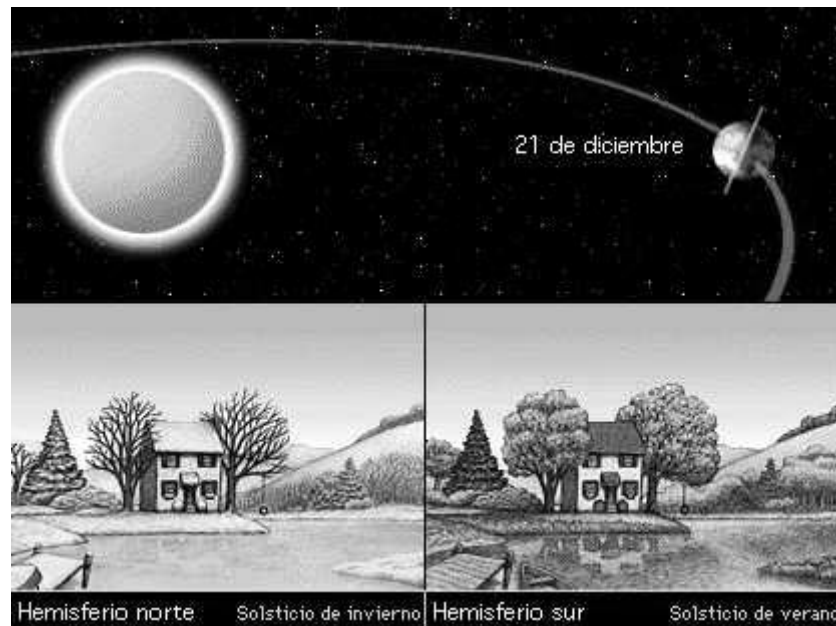


Fig. 6 Solsticio de Invierno.

En el Equinoccio de Primavera, 20 ó 21 de marzo (figura 7), los días y las noches tienen igual duración en todo el planeta, al situarse de nuevo el Sol en la vertical del Ecuador, comenzando la primavera en el Hemisferio Norte y el otoño en el Hemisferio Sur. [2]

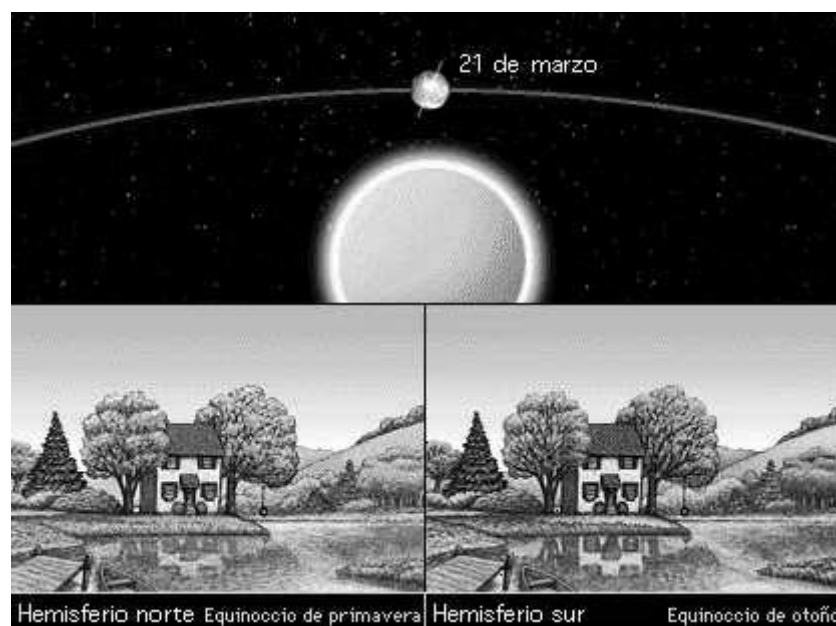


Fig. 7 Equinoccio de Primavera.

### 1.1.2. Distribución de las precipitaciones

Existen tres factores que determinan básicamente la distribución de la precipitación total anual en la Tierra: latitud, continentalidad y relieve.

El factor *latitud* se aprecia al observar el mapa en el que se representa la distribución de las precipitaciones anuales. Las *isoyetas*, líneas que unen puntos que reciben igual cantidad de precipitación, delimitan los grandes "cinturones de lluvia" de clara disposición latitudinal. [4]

La *zona ecuatorial*, bajo el dominio de la "zona de convergencia intertropical", recibe abundantes y continuas lluvias durante todo el año, más de 2.000 mm. En las *zonas tropicales húmedas* oscilan entre 2.000 y 500 mm de precipitación, disminuyendo a medida que se avanza en latitud, ya que debido al vaivén de la convergencia intertropical parte del año están bajo su influencia y parte bajo la influencia de los anticiclones tropicales. En las *zonas tropicales secas* las precipitaciones descienden progresivamente hasta ser inferiores a 250 mm anuales en los desiertos subtropicales. La cantidad de precipitación aumenta progresivamente en *latitudes medias*, donde llega a superar los 1.000 mm. Estas precipitaciones van siempre asociadas a las borrascas del frente polar. Finalmente, en las *zonas polares*, las precipitaciones descienden de nuevo hasta menos de 250 mm, debido a las masas de aire con bajo contenido en vapor de agua. [4]

La continuidad de los cinturones de lluvia de disposición latitudinal se rompe por efecto de la *distribución de mares y continentes*. De forma muy general puede decirse que el litoral recibe mayor cantidad de precipitaciones que el interior de los continentes, aunque son notables las diferencias entre unas costas y otras. [4]

La *altitud*, al menos hasta cierto nivel, acrecienta las precipitaciones, por lo que la presencia de cadenas montañosas distorsiona aún más la disposición latitudinal de las lluvias [4]. Se puede decir que una montaña constituye un enclave meteorológico y climáticamente diferenciado de las características regionales o zonales que le corresponderían.

### 1.1.3. Manifestación de la niebla

La niebla conlleva a la disminución de las condiciones de visibilidad en la superficie; generalmente la distancia visible a través de ésta es de menos de 2 km.

La niebla se clasifica según el origen del enfriamiento que dio origen la condensación, así es posible distinguir los siguientes tipos de niebla: [24]

- *Niebla de radiación*: sólo se localiza sobre la línea del suelo, iniciándose durante las noches claras, como consecuencia del enfriamiento del suelo por la radiación infrarroja del Sol. Entonces, el suelo enfriado produce la condensación del aire cercano a él, con lo que se generará una niebla que tenderá a desaparecer apenas amanece.
- *De viento*: toma lugar cuando masas de aire cargadas de humedad pasan sobre suelos fríos, lo cual enfría el aire mismo. Este fenómeno es común en las costas, cuando el tibio aire tropical se encuentra con aguas de latitudes mayores. También es común cuando un frente cálido se mueve sobre un área con una cantidad considerable de nieve.
- *Nieblas de vapor*: se da cuando aire frío se mueve sobre aguas más cálidas. El vapor del agua entra en la atmósfera por procesos de evaporación, y la condensación se da cuando se alcanza el punto de rocío. Este suceso es común en regiones polares, lagos de tamaño considerable, y al final del otoño y principio del invierno.
- *De precipitación*: en presencia de una tormenta, bien de nieve o bien de lluvia, si ésta atraviesa una masa de aire cuya temperatura es superior a la de sí misma.
- *De ladera*: se forma cuando el viento sopla contra la ladera de una montaña u otra formación geológica análoga. Al ascender en la atmósfera, la humedad se condensa. Es por esto que muchas veces las cumbres montañosas aparecen nubladas.
- *De valle*: se forma en los valles, usualmente durante el invierno. Es resultado de la inversión de temperatura, causada por aire frío que se asienta en el valle, mientras que el aire caliente pasa por encima de éste y de las montañas. Se trata básicamente de radiación confinada por un accidente orográfico, y puede durar varios días, si el clima está calmo.

### 1.2. Técnicas, algoritmos y tendencias actuales

En este epígrafe se muestran las soluciones más comunes que dan los desarrolladores de ERV a los sistemas con funcionalidades similares al sistema que se pretende desarrollar.

### 1.2.1. Iluminación

Las bases para una correcta iluminación siempre son las mismas, ya que de una forma u otra se intenta recrear la iluminación de la vida real, y ésta sigue ciertas normas y principios que no deben pasarse por alto a la hora de iluminar un entorno virtual.

#### 1.2.1.1. Modelos de Iluminación

Un modelo de iluminación representa una simulación de los fenómenos que, modificando o filtrando la distribución de energía de la luz incidente, dan lugar a los colores en las imágenes.

#### Modelo de iluminación global

El modelo de iluminación global tiene como principal característica que la luz que llega a un objeto depende no sólo de la luz directa si no de la luz que le llega desde los demás objetos de la escena [5]. La siguiente figura muestra una representación de este modelo.

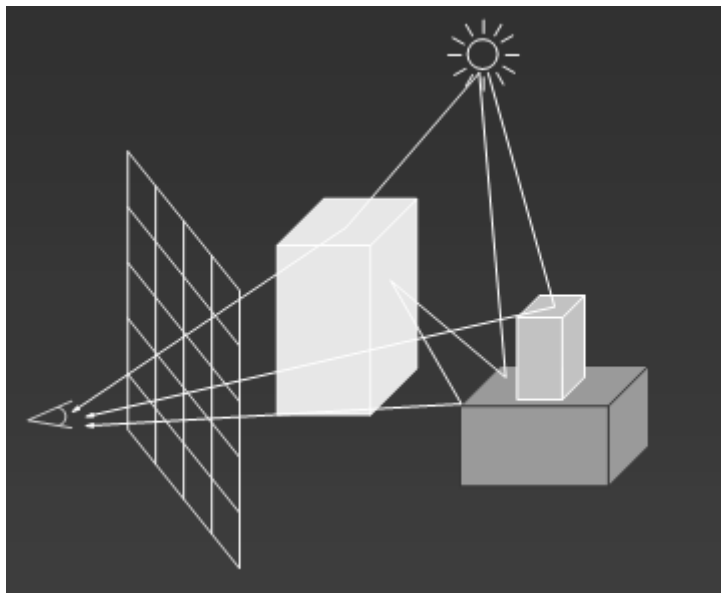


Fig. 8 Modelo de iluminación global.

Los modelos de iluminación global usan generalmente *radiosity* o *ray tracing* y son computacionalmente complejos de implementar en *real-time*. Sin embargo, si los objetos y las fuentes de luz que están en el entorno son estáticas es posible realizar los cálculos de la iluminación global como un paso de procesamiento y luego mostrar los resultados interactivamente. Este enfoque es a la vez práctico y útil para aplicaciones como paseos virtuales [6]. Entre los algoritmos de *rendering* de modelos

de iluminación global más usados están: *Ray Tracing*, *Radiosity*, *Rendering* basado en imágenes, Simulación Monte Carlo, *Photon Mapping*.

### Modelo de iluminación local

Los modelos de iluminación local consideran un punto único en una superficie y las fuentes luminosas que lo iluminan directamente, en estos modelos se omite del modelo, la interacción luminosa entre objetos y se resuelven ecuaciones a partir de la información geométrica local de un punto.

El valor de iluminación local depende tanto de las características de las luces como del material o propiedades de los objetos [7]. La siguiente figura muestra una representación de este modelo.

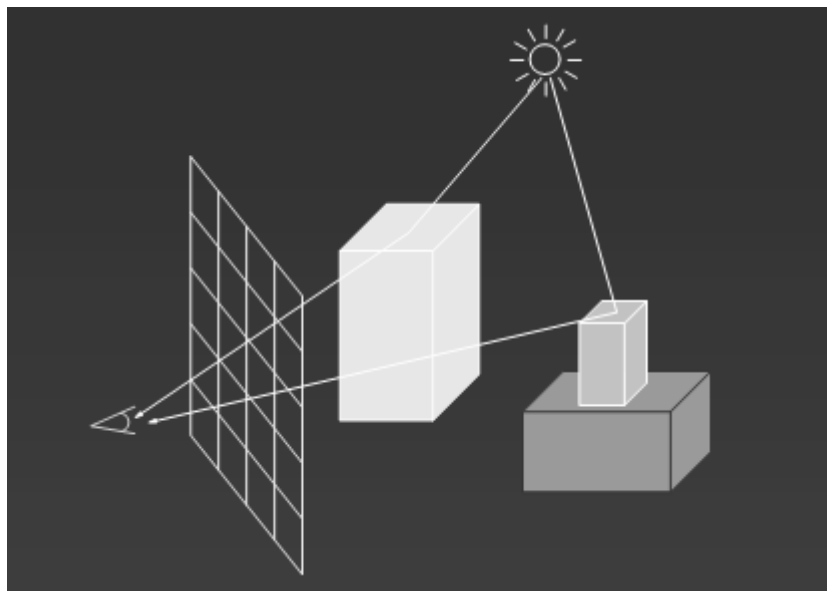


Fig. 9 Modelo de iluminación local.

Los gráficos tradicionales de los Modelos de Iluminación Local son: [8]

- Rápidos de computar.
- Heurísticos e incompletos (no físicos).
- Sobre todo interesados en la luz en dirección al ojo de observación.

Existe también otro **modelo de iluminación** llamado **Semi-Local**, el cual se basa en el lanzamiento de un rayo (*Ray Casting*), pruebas de sombra para fuentes primarias bloqueadas, fuentes de luz puntuales que producen sombras sólidas y que no haya ninguna fuente de luz secundaria en la escena. [9]

### 1.2.1.2. Tipos de Luces

Existen cuatro tipos básicos de luces: [10]

**Luz ambiental:** es la luz de fondo que ilumina igualmente todas las superficies sin importar su posición u orientación. No viene de una dirección en particular.

**Luz direccional:** fuente de luz que tiene dirección pero no posición finita. Simula, por ejemplo, los efectos de la luz solar.

**Luz punto:** fuente de luz omni-direccional (radial) que tiene posición. Se atenúa con la distancia.

**Luz Spot:** ilumina un área pequeña, con un cono de ángulo especificado (por ejemplo: linternas, luces de automóviles). Se atenúa con la distancia.

Las luces también se pueden clasificar de acuerdo a la forma en que emiten los rayos de luz: [11]

- Luz propia es proporcionada por una fuente de luz bien situada, de la cual conocemos su ubicación, y que emite sus rayos de luz de forma radial o divergente.

- Luz impropia emite sus rayos de luz desde una ubicación muy distante, de forma que podemos considerar sus rayos como paralelos entre sí.

En muchos casos estos tipos de luces se implementan de forma diferente por los desarrolladores de *engines* pero su función sigue siendo la misma.

### 1.2.1.3. Técnicas de iluminación exterior

A continuación se explica una técnica muy usada para iluminar escenas exteriores la cual cumple con los principios básicos para una correcta iluminación de una escena en un entorno 3D.

Antes que nada, se debe tener muy claro la situación ambiental que se quiere recrear, ya que las pautas que se seguirán dependen principalmente de esto. Lo primero sería definir en qué hora del día se desarrollará la escena, que estación del año, que condiciones meteorológicas, etc.

Se parte de una luz infinita que representará el Sol. Deberá ser la luz más importante de la escena y por ello la que tenga mayor intensidad. Teniendo en cuenta que la luz

del Sol no se atenúa con la distancia (realmente sí, pero se puede despreciar) por eso se elige una luz infinita (direccional) en lugar de una *spot*. Esta luz se debe teñir de un color amarillento para imitar el Sol durante la mayor parte del día. Para imitar un atardecer o un amanecer la luz se debe teñir de naranja/marrón. [12]

Si se deja la escena iluminada sólo con la luz solar se verá demasiado oscura en algunas regiones, pero esto se puede arreglar con las luces del cielo.

Añadiendo luces de cielo, se logra restar contraste a la imagen. Las sombras se teñirán ligeramente de azul para dejar de ser completamente negras y la escena ganará equilibrio de color.

Para empezar con las luces del cielo, se puede utilizar una *spot*. Se situarán los focos apuntando hacia abajo, casi verticales o con pequeños ángulos. Su intensidad no debe superar nunca al Sol [12]. Se tiñen de azul cielo y se ajustan sus parámetros hasta que al solaparse con la luz del Sol ambas luces se vea más o menos blanco.

### 1.2.2. Técnicas para simular efectos atmosféricos

#### 1.2.2.1. Usando sistemas de partículas

Algunos objetos son difíciles de representar con un conjunto de primitivas, incluso tomando la ventaja de las técnicas de transparencia y mapeado de textura. Éstos objetos poseen escasas o ninguna superficie sólida. Fenómenos naturales que tienen estas características son: el humo, las nubes, el fuego, el agua, etc.

Los sistemas de partículas pueden usarse para representar estos objetos. Un sistema de partículas es un conjunto de numerosos objetos primitivos simples, que se procesan como un grupo para representar un objeto más complejo. Las características de estos objetos como el tamaño, posición, color, y la vida de la propia partícula, pueden cambiarse de manera dinámica. Si estos parámetros son coordinados correctamente, la colección de partículas puede representar el objeto deseado. [13]

#### Representando Partículas

En la mayoría de los casos se requieren muchas partículas para crear los objetos más realistas, y casi siempre se desea realizar el procedimiento con el menor costo computacional posible. [13]

Las partículas en los sistemas de partículas se organizan en tablas (figura 10). Las partículas son puestas en un índice por partícula, conteniendo características que se actualizan en cada *frame*.

Index	X, Y, Z	R, G, B, A	Vx, Vy, Vz	Lifetime Count
0				
1				
2				
3				
...				

**Fig. 10** Tabla de partículas.

Los parámetros usados directamente para el render como los de posición (x, y, z) pueden combinarse en la tablas con los parámetros no renderizables, como la velocidad actual y el tiempo de vida. [13]

### Precipitaciones

Las precipitaciones tales como la lluvia y nieve pueden modelarse usando los sistemas de partículas descritos anteriormente. Esta tarea se desglosa en tres partes:

1. Render realista de la partícula.
2. Cálculo de la dinámica de la partícula.
3. Administración y manipulación del tiempo de vida de la partícula.

Un ejemplo son los copos de nieve; los cuales son renderizados individualmente como puntos geométricos coloreados de blanco. Las partículas deben renderizarse correctamente bajo la proyección de perspectiva y estar sujetas a los efectos de gravedad, viento, etc., los cuales deben estar incluidos en la dinámica del modelo. [13]

La dinámica de las partículas puede causar que las partículas se muevan de una porción del mundo que no es actualmente visible a la porción visible o viceversa. En el ejemplo de nieve, las partículas pueden encogerse y pueden desaparecer para que el efecto sea más realista.

Uno de los problemas más difíciles con la administración de la vida de las partículas es el fin de vida de la partícula. Normalmente los copos de nieve aumentan para formar una capa de nieve encima de los objetos en que ellos caen. Una manera de modelar esto, es terminar la dinámica de la partícula cuando la partícula golpea una superficie (usando un algoritmo de descubrimiento de colisión), siendo esta su última posición en la cual continuará dibujándose. Una dificultad con esta solución es que el



número de partículas que necesitan ser dibujadas en cada *frame* crecerá sin límite. [13]

Usar un mapa de textura para las partículas de nieve que hayan colisionado es un mecanismo eficaz para mantener un número constante de partículas en el sistema, y funciona muy bien en la simulación de la acumulación inicial de la precipitación en una superficie descubierta.

Otro método para reducir el trabajo de render y aumentar el rendimiento de la simulación es reducir el número de partículas usando una técnica basada en renderizar una “cortina” de partículas delante del espectador en vez de renderizar las partículas a lo largo del volumen entero. El uso del desenfoque de movimiento y volumen de niebla a lo largo de la iluminación para simular un cielo nublado puede hacer la ilusión más convincente. [14]

### 1.2.2.2. Cono doble

Una técnica muy usada que se traduce en alto rendimiento de simulación en tiempo real y permite la simulación de escenas estáticas y en movimiento, es la del *cono doble* (figura 11). El principio que defiende esta técnica es que en ocasiones la simulación de la lluvia y la nieve en tiempo real se hace complicada si la cadencia de estos elementos se intensifica. Esta técnica propone la texturización de un cono basada en una imagen estática extendida a lo largo de éste, lo que permite óptimas condiciones de simulación en tiempo real, manteniendo en todo momento un alto control de la simulación, y tiene en cuenta varias propiedades de la precipitación, como por ejemplo la que logra al alargar la textura para simular el desenfoque de movimiento de las partículas. Propone correr las coordenadas de la textura cíclicamente y así dar la sensación de partículas cayendo. El cono se traslada a la par de la cámara de modo que no se sale del área de emisión de las “partículas”. [15]

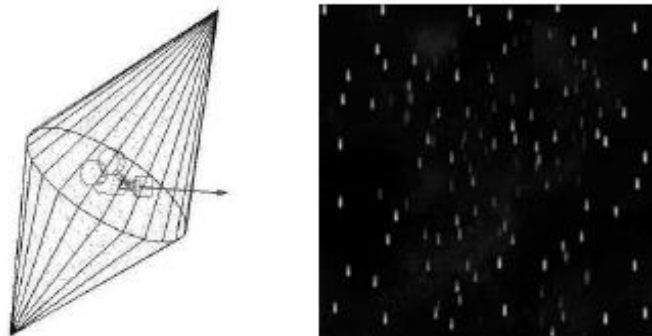


Fig. 11 Representación del cono doble y de una textura de lluvia.

### 1.2.3. Algoritmo para calcular la posición del Sol

La posición del Sol se suele expresar en coordenadas esféricas: el azimut y la altura solar, ya que son los dos ángulos que se pueden medir durante la trayectoria aparente del Sol sobre el horizonte en una localidad específica.

El ángulo que forma la dirección en la que está el Sol con la dirección norte (en el sentido de las agujas del reloj) se llama azimut. [16] y también se le puede denominar ángulo horario local el cual se puede determinar a partir de la cantidad de horas de luz del día, teniendo en cuenta que de la salida a la puesta del Sol este ángulo varía en  $180^\circ$ .

#### La altura del Sol

La altura del Sol (ver figura 12), es el ángulo que forman los rayos solares con la horizontal a la superficie terrestre ( $\gamma_s$ ). Cuanto mayor sea la altura del Sol en el cielo con respecto al cénit, más perpendicularmente incidirán los rayos y más concentrada estará la intensidad de la radiación en una determinada área de la superficie terrestre. En el sistema Tierra-atmósfera, el ángulo  $\gamma_s$  va a depender de la estación del año, de la latitud y de la hora del día. [16]

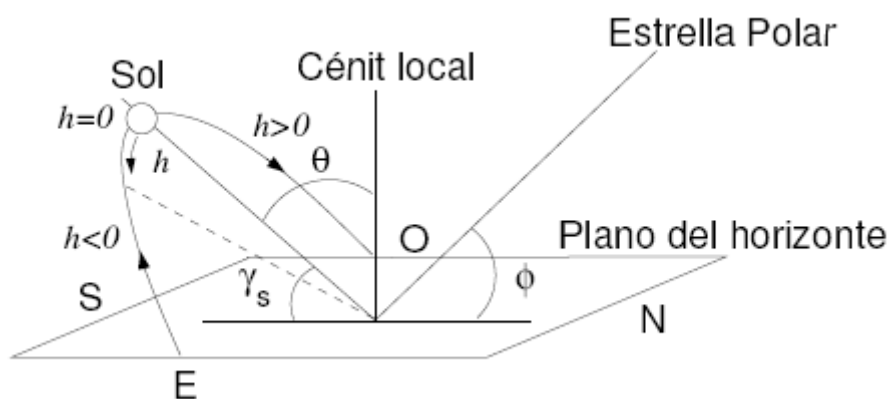


Fig. 12 Coordenadas de posicionamiento del Sol respecto de un punto en la Tierra en el hemisferio norte.

Independientemente de la latitud y hora del día, la altura del Sol viene determinada por la declinación solar  $\delta_s$ , ángulo entre la línea que conecta los centros de masa del Sol y la Tierra y el plano ecuatorial celeste (prolongación del plano del ecuador terrestre) como se muestra en la figura 13. La declinación solar se hace cero en los equinoccios de primavera y otoño, máximo ( $+23,5^\circ$ ) en el solsticio de verano y mínimo ( $-23,5^\circ$ ) en el solsticio de invierno, variando de forma casi sinusoidal el resto del año. Dado un día

$d_n$  del año (contados del 1 al 365 desde el 1 de enero), la declinación puede calcularse de forma aproximada mediante:

$$\delta_s = 23,45^\circ \cos\left(2\pi \frac{d_n - 173}{365}\right) \quad (1)$$

Siendo  $d_n = 173$  el día correspondiente al solsticio de verano (22 de junio). De la figura 13 es evidente que  $\Phi = \delta_s + \theta$ , siendo  $\Phi$  la latitud geográfica de un punto situado en la Tierra. [16]

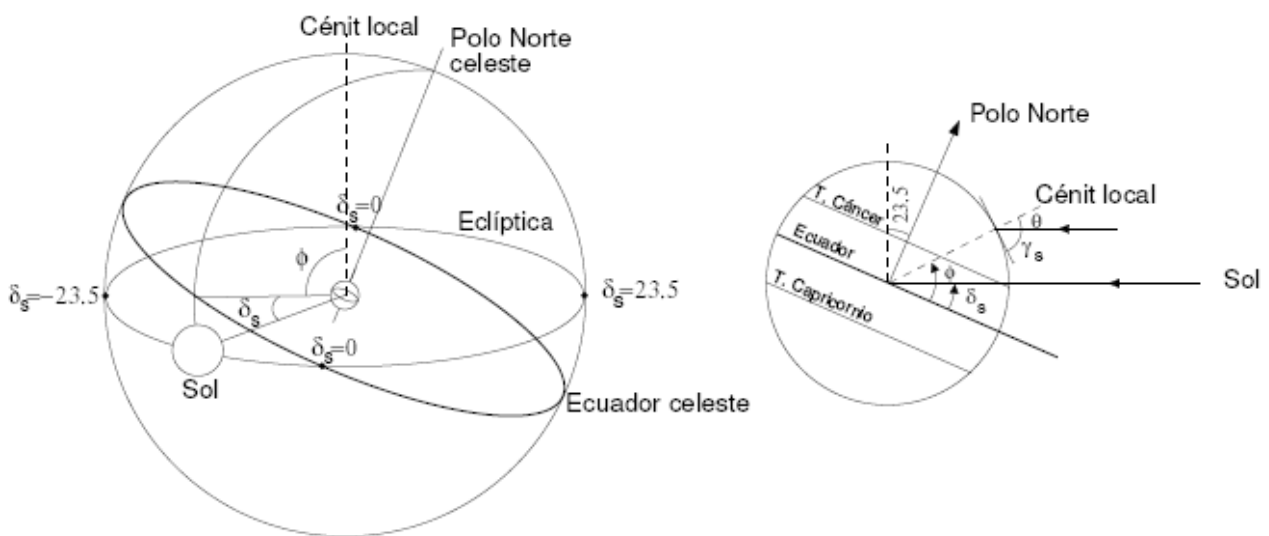


Fig. 13 Referencias celestes con las coordenadas del Sol respecto de un observador en la Tierra.

Debido a la rotación terrestre, la altura solar  $\gamma_s$ , o su complementario ángulo cenital  $\theta$ , varía a lo largo del día. Por ello, cualquier punto situado a una latitud  $\Phi$ , gira a una velocidad angular  $\omega = 2\pi / 24 \text{ rad/h}$  y ello hace que la posición aparente del Sol cambie continuamente. Para dar cuenta de este cambio se emplea el ángulo horario  $h$  u hora angular solar, que es el ángulo formado entre la posición del Sol a la hora considerada y su posición al mediodía,

$$h(t) = \frac{2\pi t}{24} - \pi \quad (2)$$

con  $t$  expresado en la hora solar local, de tal forma que  $h = 0$  a mediodía, es decir, a las 12 horas **hora local**. [16]

Llegado este punto, se puede expresar la relación angular fundamental que permitirá calcular la altura solar  $\gamma_s$  en función de la latitud geográfica  $\phi$  del punto de observación, de la declinación solar  $\delta_s$  y de la hora angular solar  $h$ : [16]

$$\sin \gamma_s = \cos \theta = \sin \delta_s \sin \phi + \cos \delta_s \cos \phi \cos h(t) \quad (3)$$

### La duración del día (horas de luz)

La expresión dada por la ecuación (3) permite calcular la duración del día si tenemos en cuenta que tanto en la puesta (ocaso) como en la salida (orto) del Sol se cumple que  $\gamma_s = 0$ . Despejando los valores de la hora angular solar  $h_0$  que verifican esta condición se obtiene que el número de horas de luz diaria:

$$\cos h_0 = -\tan \phi \tan \delta_s$$

$$|h_0| = |\arccos(-\tan \phi \tan \delta_s)| = \text{_____ radianes}$$

$$N_d = 2|h_0| \frac{180^\circ}{\pi} \frac{1}{15^\circ} = \text{_____ horas}$$

Si  $N_d$  es la longitud del día expresada en horas,  $12 - N_d/2$  representa el orto en horario solar local mientras que  $12 + N_d/2$  nos daría la hora del ocaso. [16]

### 1.2.4. El tiempo

En el mundo real toda actividad o proceso depende inevitablemente del tiempo. El tiempo controla el "ritmo" de todos los eventos que suceden en la vida diaria. Es por esta razón que resulta un atributo importante a la hora de representar un mundo virtual, y su correcta manipulación contribuirá en gran medida al realismo de una aplicación de realidad virtual.

Representar el tiempo lo más cercano a la realidad y lograr un sincronismo uniforme independientemente al CPU que procese el programa es algo a tomar en consideración. Para esto, es necesario "forzar" a que la aplicación corra siempre de la misma forma independientemente del CPU. Existen dos formas de hacer este sincronismo: en base al *framerate* (razón o frecuencia en que se muestra la animación, número de *frames* por segundo), y en base al tiempo. [17]

La sincronización en base al framerate consiste en detener el ciclo principal de simulación hasta que se desee ejecutar el siguiente ciclo. Por ejemplo, si se quiere

que el simulador corra a 24 fps., se fija el framerate y se fuerza a que cada ciclo dure 0.041 segundos. Esto se logra chequeando al final de cada ciclo si se consumió el tiempo asignado para éste, y si no ha terminado se espera hasta que se cumpla dicho tiempo. De esta manera no importa la velocidad del procesador, pues siempre se dibujarán la misma cantidad de *frames* por segundo. Éste método tiene como desventaja, que cuando se corra la aplicación en una máquina muy lenta, el ciclo podrá demorar más que el tiempo asignado para un *frame*, e igual, no existirá el sincronismo deseado.[17]

En cambio, en la sincronización en base al tiempo no importa el *framerate* que tenga la aplicación, de hecho ésta es su gran ventaja. Consiste en hacer el ciclo sin espera intermedia, y manteniendo el intervalo de tiempo transcurrido entre el ciclo anterior y el actual; con este tiempo, se calculan las nuevas posiciones de los objetos, tan sencillo como: si transcurrió poco tiempo, se moverán menos, si transcurrió un mayor tiempo, se moverán más. Es decir, hay que ajustar las distancias según el tiempo del ciclo.

Pero esto tampoco significa que sea infalible: en una computadora muy lenta, a pesar de que los objetos se sigan moviendo a la velocidad deseada, no se moverán en forma fluida sino con un parpadeo. Entre más rápida sea la computadora, el desplazamiento va a ser cada vez más fluido. Ambas técnicas tienen sus ventajas y desventajas, y escoger entre uno y el otro depende realmente de lo que se aspire con la aplicación. [17]

De manera general, para controlar el tiempo en la aplicación, hay que determinar cuántos *tics* o ciclos de aplicación hay por cada segundo de la máquina. Se obtiene entonces el tiempo actual de la máquina y se almacena en una variable que representa el tiempo de inicio de la aplicación; a partir de este valor, se llevará la referencia del tiempo durante la aplicación. [18]

### 1.3. Herramientas existentes vinculadas al campo de acción

Actualmente en el mundo existen diferentes herramientas que facilitan y optimizan la programación gráfica en ERV ya que cuentan con funcionalidades que son comúnmente usadas en la programación de aplicaciones 3D, ejemplo de estas funcionalidades son: el trabajo con animaciones, colisiones, fuerzas, modelos de agua, sistemas de partículas, entre otras. Muchas de estas herramientas facilitan aun más el trabajo de los programadores, ya que cuentan además con funcionalidades que permiten hacer representaciones del clima (lluvia, nieve, niebla, etc.), del “movimiento del Sol” y otros cuerpos celestes en tiempo real, a través de algoritmos y técnicas muy

eficientes. A continuación se darán a conocer algunas de estas herramientas y sus características principales.

### Graphics3D

Graphics3D (G3D) es un motor 3D de grado comercial con disponibilidad Open Source (Licencia BSD). Es usado en juegos, demos de tecnologías, investigaciones, simuladores militares y en el campo universitario. Provee una selección de rutinas y estructuras tan comunes que son necesitados en casi todos los programas gráficos. Simplifica bibliotecas de bajo nivel como OpenGL permitiendo que su uso sea muy fácil y sin limitaciones de funcionalidad o rendimiento. G3D brinda una base sólida, altamente optimizada para el desarrollo de aplicaciones. G3D no contiene grafo de escena, lo que permite mayor flexibilidad a la hora de estructurar las aplicaciones. [19]

G3D representa el cielo usando un cubo con una textura apropiadamente ajustada que proporciona la sensación de un entorno infinito. La clase que controla la administración del cielo puede también manipular el render de la Luna y el Sol, con efectos de “resplandor de lente” para este último. [20]

Para acceder a estas funcionalidades proporciona un grupo de funciones miembro que permiten la variación de los parámetros de iluminación del cielo que intervienen en la representación del Sol. En la mayoría de los casos reciben parámetros que especifican la hora en que comenzará la simulación.

### Ogre

OGRE, motor de renderizado 3D flexible y orientado a objetos, funciona en múltiples entornos como Windows, Mac OS X o Linux, aparte de haber sido diseñado con la idea de hacer más sencillo el desarrollo de juegos 3D que exploten al máximo las posibilidades materiales de las tarjetas 3D, y todo esto a través de una interfaz orientada a objetos. Ofrece un sistema de partículas y de gestión de recursos muy potentes, además de toda una serie de funcionalidades muy interesantes. Usa un lenguaje de descripción de materiales que permite su gestión de forma independiente al código fuente de la aplicación, brinda soporte para *vertex* y *shaders*, HLSL (DirectX), GLSL (OpenGL) y Cg (DirectX/OpenGL). Brinda soporte para texturas PNG, JPEG, BMP, DDS y DXT/S3TC, así como texturas variables en tiempo real. En cuanto al modelado, permite la edición externa desde Blender, Wings3D, 3D Studio Max, Maya. [21]

Una propiedad importante que poseen los sistemas de partículas de OGRE es *cull\_each*, en la cual todos los sistemas de partículas son truncados por una envoltura

cúbica que define las fronteras del sistema de partículas. Esta propiedad se puede aplicar a cada partícula individualmente para ganar en tiempo o a un grupo de partículas en el sistema. [22]

Debido a ésta y otras propiedades, los sistemas de partículas de OGRE se hacen muy útiles para representar efectos complejos (como precipitaciones, fluidos, humo, etc.) con un gran nivel de eficiencia, calidad y usabilidad.

# Capítulo **2**

## Soluciones Técnicas

---

---

### Introducción

En el presente capítulo inicialmente se explican los aspectos que se tendrán en cuenta para simular horarios. Seguidamente se describe cómo será el transcurso del tiempo. Se explica además cómo se realizará la simulación de las estaciones. También se describen las características que tendrán el Sol, las precipitaciones y la niebla. Aparece explicado cómo se calculará la posición del Sol y la época del año. Además, se mencionan las metodologías, herramientas y lenguajes que se usarán para el desarrollo del módulo.



### 2.1. Horarios

La simulación de horarios siempre comenzará a partir de los datos de fecha y hora en correspondencia con otros elementos que serán proporcionados por quien utilice el módulo o en dependencia de los datos obtenidos de la máquina.

Entre los principales elementos que se tendrán en cuenta para la simulación de horarios estará el redimensionado del tiempo, que se manifiesta cuando se hacen equivalencias del tiempo real para representar fracciones de tiempo, mayores o menores, en el mundo virtual. Para tratar esto, en el proceso de desarrollo se definirá una variable que contenga la magnitud de una hora de simulación en segundos, a la cual, en una aplicación final que utilice el módulo se le podrá dar el valor que se estime conveniente. Otro elemento importante que hará más precisa la simulación de horarios será la latitud, la cual además de representar la ubicación geográfica, influye de manera significativa en la duración de los horarios diurnos y nocturnos; por lo que estará presente en los datos de entrada del algoritmo que calculará la cantidad de horas diurnas y nocturnas que tendrá un día de simulación.

La época del año también influye de manera determinante en la variación de los horarios, ya que por ejemplo en el hemisferio norte, en junio, el horario diurno es más largo que el nocturno y en diciembre ocurre lo contrario. Por tanto antes de definir cualquier horario se determinará la época del año.

También se tendrá en cuenta algunos cambios ambientales relacionados con la hora del día, por ejemplo, la posición del Sol, que es diferente a cada hora del día y que servirá de guía para determinar horarios diurnos y nocturnos. Además, del horario del día, dependerá en gran medida la probabilidad de las precipitaciones y la niebla.

### 2.2. El transcurso del tiempo

Para controlar el transcurso del tiempo se utilizará la sincronización en base al tiempo, ya que independientemente del *framerate* que tenga la aplicación, hará que el tiempo en la simulación transcurra de forma invariable, de lo cual dependerá totalmente la precisión en los cálculos efectuados dentro de la simulación, ya que la mayoría de ellos se rigen por el tiempo.

El cálculo que se realizará para controlar el transcurso de las 24 horas del día y de la sucesión de los días y los meses dependerá de la magnitud de la hora, de forma tal que si ésta tiene un valor de X segundos, entonces cada X segundos transcurrirá una hora de simulación. Para este cálculo se utilizará un contador, el cual irá actualizando la hora del día, que al llegar a 24, se inicializará en cero pero el número ordinal del día

habrá cambiado. Así cuando hayan transcurrido la cantidad de días del mes que se simula (la cual estará almacenada y además habilitada para obtener este dato en cualquier momento) el contador comenzará en cero, el número ordinal del día nuevamente en 1 y el mes con su número identificador correspondiente.

### 2.3. Estaciones del año

Simular los principales fenómenos naturales que ocurren en cada época, generando y manipulando éstos de forma precisa, contribuirá en gran medida a lograr el objetivo propuesto. No se trata de exagerar con grandes efectos que distraigan al espectador, sino lograr que los efectos atmosféricos aparezcan de manera fluida, enfatizando las características que debe mostrar la época del año que se representará.

La simulación de las estaciones sugiere desde un inicio una alta complejidad, porque no basta con identificar la estación con los datos de fecha que proporciona un día; habría que determinar además la zona horaria donde se simula, y tomar en cuenta los lugares críticos como desiertos y montañas donde el clima se comporta de manera independiente a la estación vigente y a la latitud. Por estas razones se simularán las estaciones como si no existieran dichos lugares, para lograr una regularidad que permita hacer cálculos estadísticos y de probabilidad, que determinarán la ocurrencia de los efectos atmosféricos que se simularán (precipitaciones y niebla); además, de esta forma, también se reducirá el volumen de información a procesar.

Los cálculos estadísticos y de probabilidad serán basados en datos reales correspondientes a la zona climática, la estación y la hora del día; los cuales controlarán la hora de ocurrencia, tipo, intensidad, y duración de los eventos de precipitación y/o niebla.

Al inicio de cada día, en dependencia de la estación a la que pertenece, se realizarán las planificaciones de los eventos que podrán o no suceder en ese día, dicha planificación se basará en realizar inicialmente una verificación de si existen probabilidades de ocurrencia de precipitación y/o niebla en ese día, y seguidamente se definirá una cantidad de eventos organizados con sus características específicas obtenidas del cálculo probabilístico.

### 2.4. Características del Sol

Para introducir el Sol en la escena es necesario tener en cuenta algunas de sus características principales, que son indispensables para una simulación realista de éste, tales características se muestran a continuación.

### 2.4.1. Concepción general

Con el objetivo de facilitar el desarrollo de este módulo se realizará una abstracción de la realidad donde se supondrá que es el Sol el que describe una órbita alrededor de la Tierra, o sea, lo contrario a lo ocurre en la realidad. De esta forma sólo serán la posición del objeto que simule el Sol y la posición de la Luz las que cambiarán, de lo contrario si se tratara de hacer como ocurre realmente, entonces todos los objetos del mundo virtual tendrían que rotar entorno a un eje común y además habría que trasladar el mundo completo y esto evidentemente sería una solución mucho más compleja y más costosa computacionalmente.

### 2.4.2. Orientación geográfica

Con el objetivo de fomentar la usabilidad, lo más conveniente será ubicar el sistema de traslación del Sol de tal forma que coincida con los puntos cardinales del ayudante *Compass* (brújula) en 3D Studio Max, el cual sirve de referencia para la orientación de los objetos en esta herramienta. De esta forma si se diseña un mundo teniendo en cuenta la orientación que brinda este ayudante, cuando el mundo se cargue en STK, el Sol saldrá por el Este definido por el diseñador y se ocultará por el Oeste definido, haciendo que la simulación sea correcta. Así los puntos cardinales en que se basa la trayectoria del Sol quedarán de la siguiente forma:

- Norte: Y positivas.
- Sur: Y negativas.
- Este: X positivas.
- Oeste: X negativas.

### 2.4.3. Posición relativa del Sol

Normalmente cuando se viaja en un auto, en una misma dirección, y se observa el Sol, parece que se mueve a la par del auto. Para lograr este efecto se trasladará el punto que sirve como centro para calcular la órbita que describe el Sol, junto con la posición del observador, que estará representado por la cámara.

### 2.4.4. El Sol como objeto

Se modelará usando una geometría simple y tendrá una textura acorde a las características visibles del Sol: color amarillo, desenfoque radial de luz, etc. La

distancia entre la geometría y el observador podrá ser variada en dependencia de los requerimientos de la aplicación que utilice el módulo.

### 2.4.5. Color de la luz

Según la explicación del profesor Antonio Gros en su artículo [23], el Sol es una estrella amarilla, pero la luz que emite, de acuerdo a sus propiedades físicas, es percibida por el ojo humano como una luz blanca, que es el color de la luz diurna, por tanto para lograr una luz así, se usará una Luz en la que sus componentes ambiente, difusa y *specular* den como resultado una luz blanca similar a la del Sol.

### 2.4.6. Tipo de luz

Se conoce que el tamaño del Sol es muy superior al de la Tierra, y que además el Sol y la Tierra se encuentran a gran distancia; esto es la base de teoría de Eratóstenes que sustenta que los rayos del Sol llegan a la superficie terrestre prácticamente paralelos entre sí, la cual está vigente para muchos cálculos que se realizan actualmente.

Por tanto, según estas características, se usará una luz direccional debido a que como no tiene posición finita en el espacio tridimensional y sólo se considera su dirección, como se vio en el capítulo anterior, se puede deducir que en cualquier punto de una escena iluminada con este tipo de luz, los rayos de luz tendrán la misma dirección y que por lo tanto se pueden considerar paralelos entre sí, que es como se consideran realmente los rayos del Sol.

## 2.5. Características de las precipitaciones y la niebla

Cuando se simulan efectos atmosféricos como precipitaciones y niebla se debe tener mucho cuidado con el consumo de recursos y con el tiempo que demora el cálculo necesario para actualizar el estado de los efectos, ya que una mala administración de estos elementos puede descontrolar totalmente la simulación.

### 2.5.1. Precipitaciones

Para simular las precipitaciones se utilizará un procedimiento basado en la técnica del cono doble que se vio en el capítulo anterior; en éste se utilizarán planos, con el objetivo de incluir menos geometría en la escena. La textura que se utilizará será también transparente como en la técnica del cono doble y será la misma para simular cualquier tipo de precipitación; ya que estará dividida en secciones, una encima de la otra, que representarán diferentes tipos e intensidades de precipitación, tales como, el comienzo de la lluvia, llovizna, nevada intensa, terminando de nevar, etc.

En dependencia del tipo de precipitación que se quiera simular, la intensidad y el tiempo, se selecciona la sección de la textura que se mostrará. Para los casos en que comienza o termina la precipitación la sección de la textura correspondiente se mostrará sólo una vez. En los demás casos se visualizará cíclicamente, dando la sensación de continuidad al observador; esto ocurrirá hasta que se quiera interrumpir el proceso o hasta que termine el tiempo de precipitación, donde seguidamente se mostrará la sección de la textura en la que termina la precipitación.

Esta técnica se complementa con la forma en que los planos serán manejados; donde cada geometría que se utilice será una copia del plano original, es decir, se cargará un solo plano, y además esto se hará una sola vez, lo que evidentemente será un proceso eficiente. Los planos se colocarán frente a la cámara y como en la técnica del cono doble, se moverán junto con ella por la escena; además siempre estarán de frente a la dirección hacia donde esté viendo la cámara.

Se utilizarán tres planos para la representación básica de esta técnica, ya que de este modo se logrará un efecto visual aceptable y la cantidad de geometría será mínima. Para lograr el efecto de profundidad se trasladarán todos los planos a una determinada distancia y algunos un poco a los lados y así de esta forma se podrá apreciar la perspectiva a través de los planos más distantes que ya no estarán alineados a los más cercanos.

Para lograr un mayor realismo los planos se rotarán hacia la izquierda o hacia la derecha y así parecerá que la lluvia cae inclinada por causa del viento.

Los datos necesarios para simular las precipitaciones se cargarán una sola vez al iniciarse la aplicación que utilice el módulo, y se manipularán cuando se planifique cada evento de precipitación.

### 2.5.2. Niebla

Uno de los efectos que se pueden producir en cualquier estación del año es la niebla, la cual se simulará a partir de la que actualmente está implementada en *SceneToolKit*, agregando algunas funciones para controlar sus parámetros en el instante en que se requiera, para habilitarla o deshabilitarla y para definir el tiempo de duración que tendrá un evento.

### 2.5.3. Tipo, intensidad y tiempo de duración

Tanto los eventos de precipitación como los de niebla estarán definidos por varios parámetros que serán configurados a la hora de planificarlos, a través de los cuales

éstos se controlarán. En el caso de las precipitaciones los parámetros serán: tipo, intensidad y tiempo de duración, los cuales se refieren a si será lluvia o nieve, lloviznas, aguacero, poca nieve o mucha nieve y al tiempo que habrá transcurrido desde que comienza hasta que termina el evento. En los eventos de niebla serán: intensidad y tiempo de duración donde el primero se refiere a si la niebla será más espesa o menos espesa y el segundo se refiere al tiempo que la niebla estará habilitada.

### 2.6. Cálculo de la posición del Sol y la época del año

La posición del Sol en el espacio y la época del año, están muy vinculadas entre sí, tanto que la posición del Sol a cada hora cambia en dependencia de la época del año y de la latitud, por ejemplo: la posición del Sol a las 4:00 p.m. en Cuba, en el mes de diciembre, será diferente a la posición que tendrá en marzo a esa misma hora y a la vez es diferente a la posición que tendrá en cualquier otra Latitud que no sea la de Cuba.

#### 2.6.1. Posición del Sol

Para calcular la posición del Sol se usará el algoritmo descrito anteriormente en el Capítulo 1, que tendrá como entrada los siguientes datos de un día: número ordinal, mes al que pertenece, latitud y hora que se simula, y como salida dos ángulos que representarán, la altura solar y el ángulo horario o azimut, los cuales matemáticamente se pueden ver como coordenadas esféricas en el espacio, que después serán convertidas a coordenadas cartesianas, debido a que son las que se utilizan en la geometría computacional.

#### 2.6.2. Época del año

La época del año se determinará a partir de los datos de fecha contenidos en el día, comparando éstos con los intervalos de días que definirán cada época; por ejemplo, para la época de verano el intervalo será del 21 de junio al 23 de septiembre.

### 2.7. Metodología, herramientas y lenguajes

#### 2.7.1. Metodología RUP

La metodología que se utilizará en el proceso de desarrollo del módulo, es RUP, (Proceso Unificado de Desarrollo). RUP es una metodología que ha demostrado su efectividad durante muchos años, ya que numerosos proyectos la han utilizado para desarrollar *software* obteniendo resultados satisfactorios. Siguiendo los pasos

propuestos por esta metodología se obtendrá una buena documentación de la tesis así como una arquitectura robusta y flexible a futuras ampliaciones.

Las principales características que influyen en la decisión de utilizar esta metodología son:

- Centrado en arquitectura.
- Guiado por casos de uso.
- Iterativo e Incremental.
- Utilización de un único lenguaje de modelado.

### 2.7.2. Herramientas de desarrollo

#### Rational Rose

Rational Rose es una herramienta para modelado visual, que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida del desarrollo de software. Permite completar una gran parte de las disciplinas (flujos fundamentales) del RUP e incluye un conjunto de herramientas de ingeniería inversa y generación de código que allanan el camino hasta el producto final. Rational Rose ayuda a la comunicación entre los miembros del equipo, a monitorear el tiempo de desarrollo y entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común. Además, los diseñadores pueden modelar sus componentes e interfaces de forma individual y luego, unirlos con otros componentes del proyecto.

#### Microsoft Visual Studio 2005

Posee numerosas herramientas asociadas que ayudan a escribir, analizar y distribuir el código, es un compilador rápido y con muy buena detección y corrección de errores. Posee facilidad de trabajo con los elementos visuales y buena integración de estos con el código. Contiene numerosas librerías con códigos pre-escritos que ayudan en la escritura del código de la aplicación. Es perfectamente compatible con la mayoría de las librerías gráficas existentes hoy en día.

### 2.7.3. Lenguajes utilizados

#### Lenguaje de programación C++

Se escogió C++ *Standard* ya que posee características superiores a otros lenguajes. Las más importantes son: programación orientada a objetos, portabilidad, brevedad, programación modular y velocidad.

Además, se trata de un lenguaje de programación estandarizado (ISO/IEC 14882:1998), ampliamente difundido, y con una biblioteca estándar que lo ha convertido en un lenguaje universal, de propósito general, y ampliamente utilizado tanto en el ámbito profesional como en el educativo. También se seleccionó, por ser el lenguaje de programación que más dominan los desarrolladores del presente módulo.

### Lenguaje de modelado UML

Para la modelación del sistema se escogió el lenguaje UML (Unified Modeling Language, Lenguaje Unificado de Modelación). Esta decisión se debe a que se ha convertido en un estándar, ya que tiene las siguientes características:

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis y diseño, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.



# Capítulo **3**

## **Descripción de la solución propuesta.**

---

---

### **Introducción**

En el presente capítulo se describe con más detalle el módulo a desarrollar con el objetivo de tener una visión más concreta del diseño del mismo. Se definen las reglas del negocio para una mayor comprensión a la hora de implementar las funcionalidades del módulo. Queda conformado el modelo del dominio con el objetivo de dar una idea básica de la estructura del sistema. Se listan los requisitos funcionales que serán las funciones que el sistema realizará, y los requisitos no funcionales que serán las propiedades o cualidades que el producto tendrá al culminarse el ciclo de desarrollo. Además se describen los procesos que surgen al responder a las funcionalidades descritas en los requisitos funcionales (Casos de Uso).

### 3.1. Reglas del negocio

A continuación se definen las reglas que se deben tener en cuenta a la hora de desarrollar el módulo.

- El módulo de clases propuesto usará modelos 3DX para la representación de las precipitaciones y el Sol, que para ser cambiados se debe seguir las instrucciones especificadas en el **Anexo 1**.
- Los puntos cardinales especificados para el módulo deberán estar en relación con los ejes de coordenadas de la forma que se explica en el **Anexo 2**.
- La distancia del Sol al observador deberá ser mayor que la mayor distancia de un punto del mundo a otro.
- Para simular precipitaciones deberá existir al menos una cámara en la escena.
- La magnitud de una hora de simulación no deberá ser menor que 30 segundos.
- El tiempo de duración de un evento de niebla deberá ser mayor que 20 segundos.

### 3.2. Modelo del dominio

Para la representación del negocio, debido a las características del módulo, se decidió usar un Modelo del Dominio; el cual proporciona la información del modelo conceptual del sistema y brinda una idea básica del diseño real, como se muestra en la siguiente imagen.

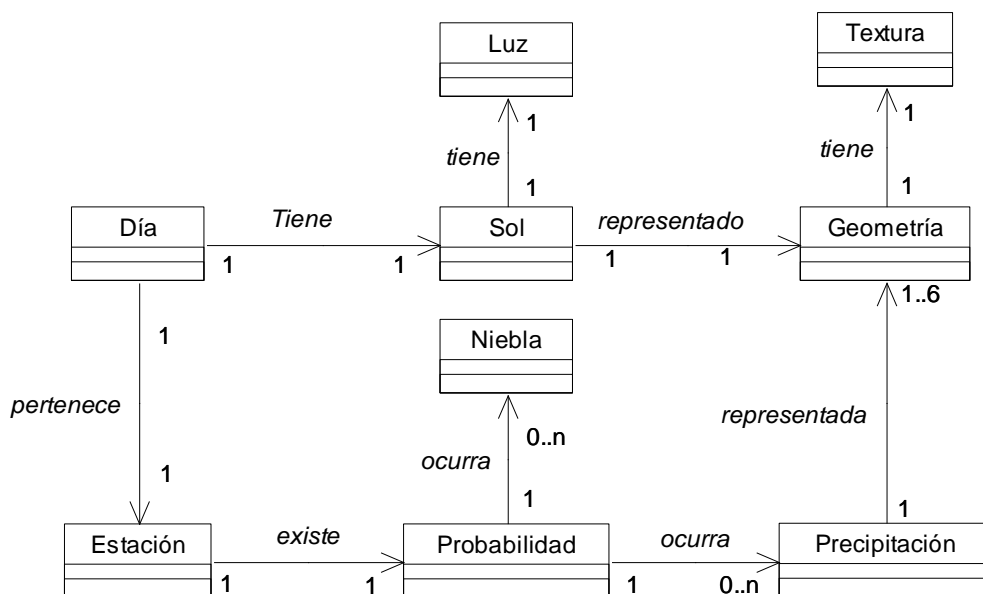


Fig. 14 Modelo del Dominio.

### 3.2.1. Glosario de Términos del Modelo del Dominio

**Día:** Está definido por las características principales que tiene un día natural (horas, fecha, Sol y condiciones climáticas).

**Estación:** Representa el comportamiento básico de las condiciones climáticas en los días.

**Luz:** Es la luz con características similares a la del Sol (color blanco, rayos perpendiculares entre sí).

**Niebla:** Estado climático donde se reducen las condiciones de visibilidad y la iluminación.

**Precipitación:** Estado climático donde se representa de forma básica las principales formas de precipitación natural (nieve y lluvia).

**Probabilidad:** Define cuándo, cómo y en qué tiempo ocurrirán los eventos de precipitación y/o niebla.

**Sol:** Es el modelo simplificado que representa las características de interés del Sol (aspecto e iluminación).

### 3.3. Captura de requisitos

A continuación se expondrán los requisitos funcionales y no funcionales del sistema.

#### 3.3.1. Requisitos Funcionales

1. Calcular la posición del Sol.
  - 1.1. Calcular horas de luz del día.
  - 1.2. Calcular altura solar.
  - 1.3. Calcular ángulo horario.
  - 1.4. Convertir a coordenadas cartesianas.
2. Variar parámetros del Sol.
  - 2.1. Cambiar geometría del Sol.
  - 2.2. Cambiar posición del Sol.
  - 2.3. Cambiar la luz del Sol.
  - 2.4. Habilitar luz del Sol.

- 2.5. Deshabilitar luz del Sol.
- 2.6. Ocultar Sol en dependencia de las condiciones climáticas.
- 2.7. Ocultar Sol (o no) en dependencia de la probabilidad de visibilidad.
- 2.8. Mostrar Sol en dependencia de las condiciones climáticas.
- 2.9. Cambiar distancia radial de la geometría del Sol al observador.
- 3. Simular precipitaciones.
  - 3.1. Crear precipitaciones.
  - 3.2. Inicializar precipitaciones.
  - 3.3. Ejecutar eventos de precipitaciones.
    - 3.3.1. En dependencia de la hora del día y la estación del año.
      - 3.3.1.1. Determinar probabilidad de que ocurran precipitaciones en un día.
      - 3.3.1.2. Determinar cantidad de eventos de precipitación que pueden ocurrir en un día así como la hora, tiempo de duración, intensidad y tipo de precipitación.
    - 3.3.2. Simular precipitación en cualquier momento sin depender de ningún proceso, sólo teniendo en cuenta el tiempo de duración, la intensidad y el tipo de precipitación.
    - 3.3.3. Iniciar los eventos de forma suavizada.
  - 3.4. Terminar eventos de precipitaciones.
    - 3.4.1. Al finalizar el tiempo planificado de un evento de precipitación.
    - 3.4.2. En el momento deseado independientemente de los demás procesos.
    - 3.4.3. Finalizar los eventos de forma suavizada.
- 4. Simular niebla.
  - 4.1. Crear niebla.
  - 4.2. Inicializar niebla.
  - 4.3. Ejecutar eventos de niebla.

### Capítulo 3: Descripción de la Solución Propuesta

- 4.3.1. En dependencia de la hora del día y la estación del año.
  - 4.3.1.1. Determinar probabilidad de que ocurra niebla en un día.
  - 4.3.1.2. Determinar cantidad de eventos de niebla que pueden ocurrir en un día así como la hora, tiempo de duración y densidad del evento.
- 4.3.2. Simular niebla en cualquier momento sin depender de ningún proceso, sólo teniendo en cuenta el tiempo de duración y la densidad del evento.
- 4.3.3. Iniciar los eventos de forma suavizada.
- 4.4. Terminar eventos de niebla.
  - 4.4.1. Al finalizar el tiempo planificado de un evento de niebla.
  - 4.4.2. En el momento deseado independientemente de los demás procesos.
  - 4.4.3. Finalizar los eventos de forma suavizada.
- 5. Configurar la magnitud de una hora de simulación.
  - 5.1. Verificar si la magnitud sea válida.
  - 5.2. Redimensionar la magnitud de una hora sin desequilibrar los demás procesos.
- 6. Simular la sucesión de los días y las noches.
  - 6.1. Definir si la simulación será o no continua.
  - 6.2. Crear día.
    - 6.2.1. A partir de datos de fecha y hora establecidos por el programador.
      - 6.2.1.1. En función de una hora real.
      - 6.2.1.2. En función de una hora personalizada.
    - 6.2.2. A partir de datos de fecha y hora obtenidos de la máquina.
      - 6.2.1.1. En función de una hora real.
      - 6.2.1.2. En función de una hora personalizada.
  - 6.3. Mantener constante la distancia entre el Sol y el observador.
- 7. Simular cada estación del año.

7.1. Identificar estaciones del año.

7.1.1. Comparar la fecha del día que se simula con el rango de fecha de las estaciones.

7.1.2. Ubicar el hemisferio donde se efectúa la simulación a través de la latitud establecida.

7.2. Simular clima de cada estación.

### 3.3.2. Requisitos no Funcionales

**Hardware:** Se necesitará un dispositivo gráfico (tarjeta gráfica) de rendimiento medio que sea capaz de procesar con rapidez y eficiencia los efectos visuales. Tarjeta de memoria RAM de 256 MB en adelante.

**Rendimiento:** Debido a su utilización en aplicaciones en tiempo real debe alcanzar una velocidad óptima de lectura de ficheros, cálculos y visualización de los elementos en la escena.

**Usabilidad:** Los usuarios de este módulo serán programadores con conocimientos elementales de programación, y el módulo les permitirá centrarse en un proyecto específico sin tener que preocuparse en los efectos climáticos que éste pueda requerir.

**Soporte:** Deberá ser multiplataforma, perfectamente compatible a las distribuciones de Linux y versiones de Windows.

**Diseño e implementación:** Proporcionará una interfaz de acceso a todas las funciones requeridas; que hará que todas las implementaciones funcionales del módulo sean transparentes a quien lo utilice. El módulo se programará completamente usando lenguaje *C++ Standard* guiado en la filosofía de la Programación Orientada a Objetos (POO).

**Software:** Sistema operativo Windows y distribuciones de Linux.

### 3.4. Modelo de casos de usos del sistema

En esta sección, se conciben los casos de uso del sistema así como los actores que van a interactuar con ellos. Además, se seleccionan los casos de uso correspondientes al ciclo de desarrollo para hacerles sus descripciones textuales en formato expandido.

### 3.4.1. Actores del sistema

Actores	Justificación
Aplicación Final	Es quien hará uso y se beneficiará con las funcionalidades que brinda el módulo de clases.

Tabla 1 Descripción del actor del sistema.

### 3.4.2. Casos de uso del sistema (CUS)

- 1- Crear Día.
- 2- Cambiar Magnitud Hora.
- 3- Configurar Parámetros del Sol.
- 4- Gestionar Precipitación.
- 5- Actualizar Día.
- 6- Actualizar Sol.
- 7- Actualizar Precipitación.
- 8- Actualizar Niebla.
- 9- Calcular Probabilidad de Precipitación o Niebla.
- 10- Crear Estación.
- 11- Actualizar Estación.
- 12- Gestionar Niebla.
- 13- Hacer Ciclo de Escena (STK).

3.4.3. Diagrama de casos de uso del sistema

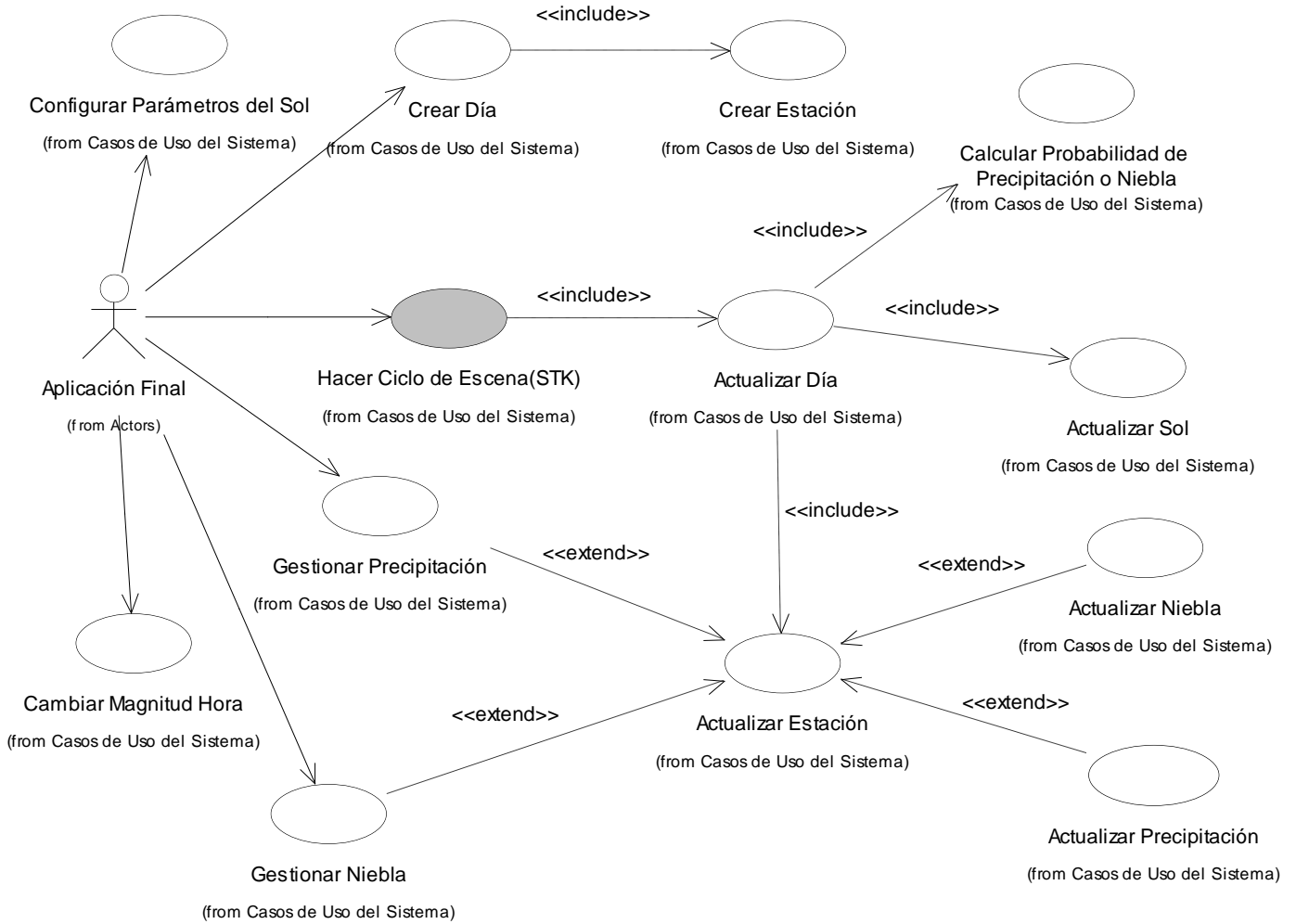


Fig. 15 Diagrama de casos de uso del sistema.

3.4.4. Especificación de los CUS en formato expandido

A continuación se muestra la especificación de los casos de uso que representan cada proceso a desarrollar en el sistema.

Al caso de uso “Hacer Ciclo de Escena” no se le realizará la especificación, ya que está implementado en la herramienta STK, de la cual formará parte este módulo. Dicho caso de uso aparece en el modelo de CUS porque se relaciona de forma directa con los procesos del sistema que se pretende desarrollar.



### Capítulo 3: Descripción de la Solución Propuesta

<b>Nombre del CU</b>	Crear día	
<b>Actores</b>	Aplicación Final	
<b>Propósito</b>	Inicializar los datos necesarios para la simulación de un día.	
<b>Resumen</b>	La Aplicación final crea un día donde se inicializan todos los parámetros requeridos para la simulación de un día.	
<b>Referencias</b>	RF6	
<b>Precondiciones</b>	-	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>	
<p>1. Inicia el caso de uso definiendo una latitud para la creación de un día de una de las siguientes formas:</p> <p>a) Con datos obtenidos de la máquina, magnitud de una hora igual a 60 minutos, y definiendo si la simulación será continua.</p> <p>b) Con datos obtenidos de la máquina y magnitud de una hora de modo personalizado.</p> <p>c) Con datos establecidos por el programador, magnitud de una hora igual a 60 minutos y definiendo si la simulación será continua.</p> <p>d) Con datos establecidos por el programador y magnitud de una hora de modo personalizado.</p>		
<b>Sección 1: Crear día (a)</b>		
1. Solicita crear un día con las características a).	<p>1.1. Captura y almacena la fecha y hora de la máquina.</p> <p>1.2. Convierte el dato de hora obtenido a formato <b>float</b>.</p> <p>1.3. Crea un día con la latitud especificada y los datos de fecha obtenidos, y dentro de éste se crea el Sol, la estación y la probabilidad.</p> <p>1.4. Establece la hora para el inicio de la simulación, que será el valor de hora procesado anteriormente.</p> <p>1.5. Habilita o no la simulación continua en dependencia de lo especificado.</p> <p>1.6. Establece la magnitud de una hora igual a 60 minutos.</p>	
<b>Sección 2: Crear día (b)</b>		
1. Solicita crear un día con las características b).	<p>1.1. Captura y almacena la fecha y hora de la máquina.</p> <p>1.2. Convierte el dato de hora obtenido a formato <b>float</b>.</p> <p>1.3. Verifica que la magnitud definida para la hora sea mayor que 30 segundos, en caso contrario establece este valor.</p>	

### Capítulo 3: Descripción de la Solución Propuesta

	<p>1.7. Crea un día con la latitud especificada y los datos de fecha obtenidos, y dentro de éste se crea el Sol, la estación y la probabilidad.</p> <p>1.4. Establece la hora para el inicio de la simulación, que será el valor de hora procesado anteriormente.</p> <p>1.5. Habilita la simulación continua.</p> <p>1.6. Establece la magnitud de una hora igual al valor definido.</p>
<b>Sección 3: Crear día (c)</b>	
1. Solicita crear un día con las características c).	<p>1.1. Verifica que la magnitud definida para la hora sea mayor que 30 segundos, en caso contrario establece este valor.</p> <p>1.2. Crea un día con la latitud y los datos de fecha definidos, y dentro de este se crea el Sol, la estación y la probabilidad.</p> <p>1.3. Establece la hora para el inicio de la simulación, que será la hora definida al inicio.</p> <p>1.4. Habilita la simulación continua.</p> <p>1.5. Establece la magnitud de una hora igual al valor definido.</p>
<b>Sección 4: Crear día (d)</b>	
1. Solicita crear un día con las características d).	<p>1.1. Crea un día con la latitud y los datos de fecha definidos, y dentro de éste se crea el Sol, la estación y la probabilidad.</p> <p>1.2. Establece la hora para el inicio de la simulación, que será la hora definida al inicio.</p> <p>1.3. Habilita o no la simulación continua en dependencia de lo especificado.</p> <p>1.4. Establece como magnitud de una hora 60 minutos.</p>
<b>Flujo alternativo</b>	
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>
<b>Poscondiciones</b>	-
<b>Prioridad</b>	Crítico

Tabla 2 Descripción del CU Crear día.

<b>Nombre del CU</b>	Cambiar Magnitud Hora
<b>Actores</b>	Aplicación Final.
<b>Propósito</b>	Establecer un nuevo valor para el transcurso del tiempo en la simulación del día.
<b>Resumen</b>	La Aplicación Final establece el nuevo valor que tendrá la magnitud de cada hora dentro de la simulación del día.
<b>Referencias</b>	RF5
<b>Precondiciones</b>	El día debe haberse creado.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>
1. Solicita cambiar la magnitud de una hora de simulación.	<p>1.1. Verifica la validez de los datos entrantes.</p> <p>1.2. Cambia el valor existente por el valor</p>

### Capítulo 3: Descripción de la Solución Propuesta

	establecido.
<b>Flujo alternativo</b>	
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>
<b>Poscondiciones</b>	El sistema simulará el día basado en la nueva magnitud de la hora.
<b>Prioridad</b>	Secundario

Tabla 3 Descripción del CU Cambiar Magnitud Hora.

<b>Nombre del CU</b>	Configurar Parámetros del Sol	
<b>Actores</b>	Aplicación Final	
<b>Propósito</b>	Modificar los parámetros del Sol en el momento deseado.	
<b>Resumen</b>	La Aplicación Final solicita configurar parámetros del Sol y se establecen los datos deseados.	
<b>Referencias</b>	RF2	
<b>Precondiciones</b>	El día debe haber sido creado y con éste el Sol.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>	
1. Solicita modificar los parámetros del Sol de las siguientes formas: a) Cambiar geometría. b) Cambiar luz. c) Habilitar/Deshabilitar luz. d) Cambiar la distancia Sol-Observador.		
<b>Sección 1: Modificar parámetro (a)</b>		
1. Solicita cambiar la geometría que representa el Sol, por la que está en la dirección que especifica.	1.1. Carga y almacena la nueva geometría correspondiente a la dirección indicada. 1.2. Sustituye la geometría anterior por la nueva.	
<b>Sección 2: Modificar parámetro (b)</b>		
1. Solicita cambiar la luz del Sol, por otra específica.	1.1. Reemplaza la luz anterior por la creada para este objetivo.	
<b>Sección 3: Modificar parámetro (c)</b>		
1. Solicita habilitar o deshabilitar la luz del Sol, según se requiera.	1.1. Habilita o deshabilita la luz del Sol en dependencia de la instrucción dada.	
<b>Sección 4: Modificar parámetro (d)</b>		
1. Solicita cambiar la distancia existente entre el Sol y el punto donde se encuentra el observador.	1.1. Verifica el valor de la distancia entrada. 1.2. Establece la nueva distancia radial.	
<b>Flujo alternativo</b>		
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>	
<b>Poscondiciones</b>	-	
<b>Prioridad</b>	Secundario	

Tabla 4 Descripción del CU Configurar Parámetros del Sol.

### Capítulo 3: Descripción de la Solución Propuesta

<b>Nombre del CU</b>	Gestionar Precipitación	
<b>Actores</b>	Aplicación Final	
<b>Propósito</b>	Gestionar las operaciones relacionadas con un evento de precipitación.	
<b>Resumen</b>	La Aplicación Final o el sistema pueden inicializar, ejecutar, o detener un evento de precipitación en el momento deseado.	
<b>Referencias</b>	RF3	
<b>Precondiciones</b>	El día debe haber sido creado.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>	
1. La Aplicación Final o el sistema solicita realizar las siguientes acciones referentes a las precipitaciones. a) Inicializar Precipitación. b) Ejecutar Precipitación. c) Parar Precipitación.		
<b>Sección 1: Acción (a)</b>		
1. Solicita inicializar un evento de precipitación definiendo tiempo de duración del evento, tipo, e intensidad.	a. Inicializa la precipitación respondiendo a los parámetros solicitados.	
<b>Sección 2: Acción (b)</b>		
1. Solicita la ejecución del evento.	1.1. Realiza la ejecución del evento de precipitación es ejecutado.	
<b>Sección 3: Acción (c)</b>		
1. Solicita detener la precipitación.	1.1. Hace que disminuya la precipitación de manera suavizada hasta detenerla totalmente.	
<b>Flujo alternativo</b>		
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>	
<b>Poscondiciones</b>	-	
<b>Prioridad</b>	Crítico	

Tabla 5 Descripción del CU Gestionar Precipitación.

<b>Nombre del CU</b>	Actualizar Día	
<b>Actores</b>		
<b>Propósito</b>	Actualizar constantemente los datos que garantizan la simulación de un día.	
<b>Resumen</b>	Se actualizan constantemente las variables que intervienen en la simulación del día.	
<b>Referencias</b>	RF1, RF2.2, RF6.3, RF2.6, RF2.7, RF2.8, RF7, incluye CUS6, CUS9 y CUS11.	
<b>Precondiciones</b>	El día debe haberse creado.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>	

### Capítulo 3: Descripción de la Solución Propuesta

1. El ciclo de escena de STK solicita la actualización constante de los parámetros del día.	<p>a. Calcula constantemente la posición del Sol, basada en la altura solar y ángulo horario.</p> <p>b. Actualiza la distancia del Sol al observador.</p> <p>c. Verifica si es de noche para ocultar el Sol.</p> <p>d. Si es de día y se inicia una precipitación oculta el Sol en dependencia de la probabilidad de visibilidad.</p> <p>e. Muestra el Sol al terminar un evento de precipitación.</p> <p>f. Actualiza la estación correspondiente.</p>
<b>Flujo alternativo</b>	
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>
<b>Poscondiciones</b>	-
<b>Prioridad</b>	Crítico

Tabla 6 Descripción del CU Actualizar Día.

<b>Nombre del CU</b>	Actualizar Sol
<b>Actores</b>	
<b>Propósito</b>	Actualizar los parámetros del Sol en cada instante.
<b>Resumen</b>	Se calcula y actualiza constantemente la posición del Sol, la distancia al centro y se visualiza o no la geometría de éste.
<b>Referencias</b>	RF2
<b>Precondiciones</b>	El Sol debe haberse creado.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>
1. La actualización del día solicita constantemente actualizar los parámetros del Sol.	<p>1.1. Actualiza constantemente la posición del Sol.</p> <p>1.2. Actualiza los parámetros del Sol en caso de ser cambiada la distancia, la geometría o la luz de éste.</p>
<b>Flujo alternativo</b>	
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>
<b>Poscondiciones</b>	-
<b>Prioridad</b>	Crítico

Tabla 7 Descripción del CU Actualizar Sol.

<b>Nombre del CU</b>	Actualizar Precipitación
<b>Actores</b>	
<b>Propósito</b>	Garantizar la simulación de los eventos de precipitación.
<b>Resumen</b>	Actualiza constantemente las variaciones en los eventos de precipitación.
<b>Referencias</b>	RF3
<b>Precondiciones</b>	Debe haberse creado un evento de Precipitación.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>

### Capítulo 3: Descripción de la Solución Propuesta

1. La actualización de la estación requiere que la precipitación se actualice siempre que esté activa.	1.1. Verifica el tipo de precipitación activa. 1.2. Muestra repetidamente la sección correspondiente a ese evento dentro de la textura. 1.3. Verifica el tiempo de duración del evento para cuando llegue el final de éste mostrar la sección correspondiente a la terminación del evento.
<b>Flujo alternativo</b>	
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>
<b>Poscondiciones</b>	-
<b>Prioridad</b>	Crítico

Tabla 8 Descripción del CU Actualizar Precipitación.

<b>Nombre del CU</b>	Actualizar Niebla
<b>Actores</b>	
<b>Propósito</b>	Garantizar la simulación de los eventos de niebla.
<b>Resumen</b>	Actualiza constantemente las variaciones en los eventos de niebla.
<b>Referencias</b>	RF4
<b>Precondiciones</b>	El estado de niebla de STK debe haberse habilitado.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>
1. La actualización de la estación requiere que la niebla se actualice siempre que esté activa.	1.1. Cuando se requiera, habilita o deshabilita el estado de niebla de STK. 1.2. Cuando se habilita el estado de niebla, aumenta la densidad de manera suavizada hasta llegar al máximo valor definido. 1.3. Al terminar un evento de niebla, disminuye la densidad de manera suavizada hasta que se deshabilita el estado. 1.4. En caso de ser solicitado, realiza variaciones en los parámetros de densidad de la niebla de STK.
<b>Flujo alternativo</b>	
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>
<b>Poscondiciones</b>	-
<b>Prioridad</b>	Crítico

Tabla 9 Descripción del CU Actualizar Niebla.

### Capítulo 3: Descripción de la Solución Propuesta

<b>Nombre del CU</b>	Calcular Probabilidad de Precipitación o Niebla	
<b>Actores</b>		
<b>Propósito</b>	Calcular la probabilidad de <b>ocurrencia</b> y de <b>tipo</b> de los eventos de precipitación y/o niebla en un día.	
<b>Resumen</b>	Calcula una secuencia de probabilidades que determina cuando ocurrirán y de qué forma, los eventos de precipitación y/o niebla.	
<b>Referencias</b>	RF3.3.1.1, RF3.3.1.2, RF4.3.1.1, RF4.3.1.2	
<b>Precondiciones</b>	El día debe haberse creado.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>	
1. La actualización del día solicita al comenzar cada día calcular las probabilidades de precipitación y niebla	1.1. Calcula la probabilidad de que se produzcan o no precipitaciones y/o niebla en un día. 1.2. Calcula el número de eventos de precipitaciones y/o niebla que tendrá un día. 1.3. Para cada evento de precipitación y/o niebla calcula la hora, la intensidad, y la duración y además define el tipo (lluvia o nieve) en el caso de las precipitaciones.	
<b>Flujo alternativo</b>		
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>	
<b>Poscondiciones</b>	-	
<b>Prioridad</b>	Crítico	

Tabla 10 Descripción del CU Calcular Probabilidad de Precipitación o Niebla.

<b>Nombre del CU</b>	Crear Estación	
<b>Actores</b>		
<b>Propósito</b>	Crear los elementos necesarios para la simulación de una estación.	
<b>Resumen</b>	Se crean los eventos de precipitación y/o niebla que se utilizarán en la simulación.	
<b>Referencias</b>	RF7	
<b>Precondiciones</b>	El día debe haberse creado.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>	
1. La creación del día solicita la creación de una estación.	1.1. Crea los eventos de precipitación con sus características por defecto. 1.2. Crea los eventos de niebla con sus características por defecto.	
<b>Flujo alternativo</b>		
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>	
<b>Poscondiciones</b>	-	
<b>Prioridad</b>	Crítico	

Tabla 11 Descripción del CU Crear Estación.

### Capítulo 3: Descripción de la Solución Propuesta

<b>Nombre del CU</b>	Actualizar Estación	
<b>Actores</b>		
<b>Propósito</b>	Actualizar constantemente las variaciones de los parámetros en los eventos de precipitaciones y/o niebla.	
<b>Resumen</b>	Actualiza, cuando se requiera, los parámetros en los eventos de precipitación y/o niebla.	
<b>Referencias</b>	RF7, extiende CUS4, CUS7, CUS8 y CUS13.	
<b>Precondiciones</b>	La estación debe haberse creado.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>	
1. La actualización de día requiere la constante actualización de la estación para el control de los eventos.	En caso de que existan eventos de precipitación y/o niebla: 1.1. Actualiza, en dependencia de la estación del año que se simula, los parámetros en los eventos de precipitación. 1.2. Actualiza, en dependencia de la estación del año que se simula, los parámetros en los eventos de niebla.	
<b>Flujo alternativo</b>		
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>	
<b>Poscondiciones</b>	-	
<b>Prioridad</b>	Crítico	

Tabla 12 Descripción del CU Actualizar Estación.

<b>Nombre del CU</b>	Gestionar Niebla	
<b>Actores</b>	Aplicación Final	
<b>Propósito</b>	Gestionar las operaciones relacionadas con un evento de niebla.	
<b>Resumen</b>	La Aplicación Final o el sistema pueden inicializar, ejecutar, o detener un evento de niebla en el momento deseado.	
<b>Referencias</b>	RF4	
<b>Precondiciones</b>	El día debe haber sido creado.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>	
1. La Aplicación Final o el sistema solicita realizar las siguientes acciones referentes a la niebla. a) Inicializar niebla. b) Ejecutar niebla. c) Detener niebla.		
<b>Sección 1: Acción (a)</b>		
1. Solicita inicializar un evento de niebla definiendo el tiempo de duración del evento y la densidad.	1.1. Inicializa el evento respondiendo a los parámetros solicitados.	
<b>Sección 2: Acción (b)</b>		
1. Solicita la ejecución del evento.	1.1. Ejecuta el evento de niebla.	



### Capítulo 3: Descripción de la Solución Propuesta

<b>Sección 3: Acción (c)</b>	
1. Solicita detener el evento.	1.1. Detiene el evento deshabilitando el estado de niebla.
<b>Flujo alternativo</b>	
<b>Acciones del Actor</b>	<b>Respuestas del Sistema</b>
<b>Poscondiciones</b>	-
<b>Prioridad</b>	Crítico

Tabla 13 Descripción del CU Gestionar Niebla.

# Capítulo **4**

## **Diseño e Implementación del Sistema**

---

### **Introducción**

En este capítulo se muestra el diagrama de clases de diseño del sistema, obtenido como resultado del refinamiento de las etapas anteriores. Se describen detalladamente las clases y sus funcionalidades. Seguidamente se muestran los diagramas de secuencia, elaborados a partir de los casos de uso que intervienen en el desarrollo del módulo. Además se explica el estándar de codificación usado y se muestran los diagramas de despliegue y de componentes del módulo.

### 4.1. Diagrama de clases de diseño

Antes de pasar a mostrar el diagrama de clases, se deben tener en cuenta algunas cuestiones importantes para una mejor comprensión de éste:

1. Para no extender innecesariamente el diagrama de clases se obviaron los tradicionales “*gets*” y “*sets*” así como los destructores.
2. No aparecen representadas las clases que se reutilizan de la herramienta STK.
3. La nomenclatura utilizada en el diagrama se explica en el epígrafe “Estándares de codificación” el capítulo posterior.

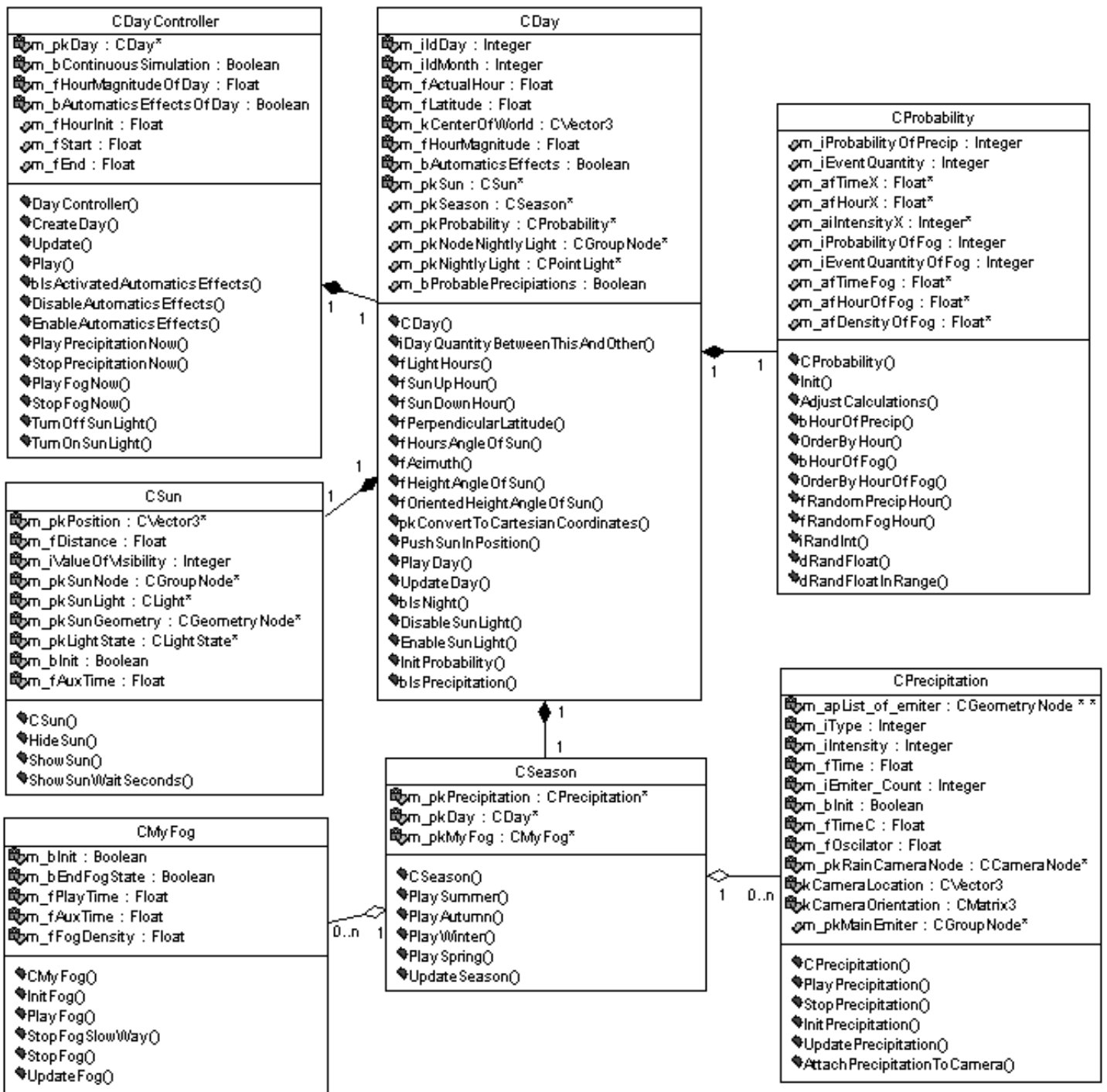


Fig. 16 Diagrama de clases de diseño.

## 4.2. Descripción de las clases del diseño

A continuación se describen las clases del diseño del módulo. Los métodos de acceso a miembros de clases (set y get) no son especificados en las descripciones, ni tampoco los destructores, puesto que su funcionalidad es sencilla. Las clases del STK que se utilizan no se describen, ya que sólo son reutilizadas en el módulo.

<b>Nombre</b>	<b><i>CDayController</i></b>	
<b>Tipo de clase</b>	Interfaz	
<b>Atributo</b>		<b>Tipo</b>
<i>m_pkDay</i>		<i>CDia*</i>
<i>m_bContinuousSimulation</i>		<i>bool</i>
<i>m_fHourMagnitudeOfDay</i>		<i>float</i>
<i>m_bAutomaticsEffectsOfDay</i>		<i>bool</i>
<i>m_fHourInit</i>		<i>float</i>
<i>m_fStart</i>		<i>float</i>
<i>m_fEnd</i>		<i>float</i>
<b>Para cada responsabilidad</b>		
<b>Nombre</b>	<b><i>CDayController</i></b>	
<b>Descripción</b>	Constructor de la clase.	
<b>Nombre</b>	<b><i>CreateDay</i></b>	
<b>Descripción</b>	Se encarga de crear un día. Se pasan como parámetros los identificadores de la fecha (día y mes), latitud y la hora donde comenzará la simulación, así como la magnitud de una hora de simulación. Establece por defecto que la simulación será continua.	
<b>Nombre</b>	<b><i>CreateDay</i></b>	
<b>Descripción</b>	Se encarga de crear un día. Se pasan como parámetros los identificadores de la fecha (día y mes), latitud y la hora donde comenzará la simulación. Además se define por parámetros si la simulación será continua o estática. Define la magnitud de una hora de simulación igual a una hora real.	
<b>Nombre</b>	<b><i>CreateDay</i></b>	
<b>Descripción</b>	Se encarga de crear un día. Se pasan como parámetros la latitud y la magnitud de una hora de simulación. Obtiene los datos de fecha y hora de la máquina. Establece por defecto que la simulación será continua.	
<b>Nombre</b>	<b><i>CreateDay</i></b>	
<b>Descripción</b>	Se encarga de crear un día. Se pasan como parámetros la latitud y si la simulación será continua o estática. Obtiene los datos de fecha y hora de la máquina. Define la magnitud de una hora de simulación igual a una hora real.	
<b>Nombre</b>	<b><i>Update</i></b>	
<b>Descripción</b>	Verifica si la simulación es continua para actualizar la hora actual y seguidamente actualiza el día.	
<b>Nombre</b>	<b><i>Play</i></b>	
<b>Descripción</b>	Verifica si la simulación es continua, para cambiar la fecha al día siguiente cuando hayan transcurrido 24 horas de simulación, inicia el cálculo de probabilidad de precipitaciones y/o niebla a la hora <b>ceró</b> de cada día y además inicia la ejecución del día. En caso de que la simulación no sea continua, no se cambia el día ni la hora.	
<b>Nombre</b>	<b><i>blsActivatedAutomaticsEffects</i></b>	

<b>Descripción</b>	Verifica el estado del atributo <i>m_bAutomaticsEffectsOfDay</i> que define si está o no activada la simulación automática de los efectos de precipitaciones y/o niebla.
<b>Nombre</b>	<b><i>DisableAutomaticsEffects</i></b>
<b>Descripción</b>	Deshabilita la simulación automática de los efectos de precipitaciones y/o niebla.
<b>Nombre</b>	<b><i>EnableAutomaticsEffects</i></b>
<b>Descripción</b>	Habilita la simulación automática de los efectos de precipitaciones y/o niebla.
<b>Nombre</b>	<b><i>PlayPrecipitationNow</i></b>
<b>Descripción</b>	Se pasan como parámetros el tipo, el tiempo de duración y la intensidad de la precipitación a ejecutar. Ejecuta un evento de precipitación con las características definidas.
<b>Nombre</b>	<b><i>StopPrecipitationNow</i></b>
<b>Descripción</b>	Detiene los eventos de precipitación.
<b>Nombre</b>	<b><i>PlayFogNow</i></b>
<b>Descripción</b>	Se pasan como parámetros la densidad y el tiempo de duración de un evento de niebla que es ejecutado de acuerdo a estas características.
<b>Nombre</b>	<b><i>StopFogNow</i></b>
<b>Descripción</b>	Detiene los eventos de niebla.
<b>Nombre</b>	<b><i>TurnOffSunLight</i></b>
<b>Descripción</b>	Deshabilita la luz del Sol.
<b>Nombre</b>	<b><i>TurnOnSunLight</i></b>
<b>Descripción</b>	Habilita la luz del Sol.

Tabla 14 Descripción de la clase del diseño "CDayController".

	<b>CDay</b>
<b>Tipo de clase</b>	Control
<b>Atributo</b>	<b>Tipo</b>
<i>m_pkSun</i>	<i>CSun*</i>
<i>m_ildDay</i>	<i>int</i>
<i>m_ildMonth</i>	<i>int</i>
<i>m_fActualHour</i>	<i>float</i>
<i>m_fLatitude</i>	<i>float</i>
<i>m_kCenterOfWorld</i>	<i>CVector3</i>
<i>m_fHourMagnitude</i>	<i>float</i>
<i>m_bAutomaticsEffects</i>	<i>bool</i>
<i>m_pkSeason</i>	<i>CSeason*</i>
<i>m_pkProbability</i>	<i>CProbability*</i>
<b>Para cada responsabilidad</b>	
<b>Nombre</b>	<b>CDay</b>
<b>Descripción</b>	Constructor de la clase.
<b>Nombre</b>	<b>CDay</b>

## Capítulo 4: Diseño e Implementación del Sistema

<b>Descripción</b>	Constructor de la clase. Se pasan por parámetros los identificadores del día, mes y latitud. Crea el Sol y establece como centro el vector nulo para la simulación del movimiento de éste. Además crea una estación y las probabilidades de precipitación y/o niebla que tendrá el día.
<b>Nombre</b>	<b><i>iDayQuantityBetweenThisAndOther</i></b>
<b>Descripción</b>	Calcula la cantidad de días en forma ordinal entre el día actual y otro día cualquiera.
<b>Nombre</b>	<b><i>fLightHours</i></b>
<b>Descripción</b>	Calcula la cantidad de horas de luz que tendrá el día.
<b>Nombre</b>	<b><i>fSunUpHour</i></b>
<b>Descripción</b>	Se pasa como parámetro las horas de luz del día y a partir de este dato calcula la hora de salida del Sol.
<b>Nombre</b>	<b><i>fSunDownHour</i></b>
<b>Descripción</b>	Se pasa como parámetro las horas de luz del día y a partir de este dato calcula la hora en que se pone el Sol.
<b>Nombre</b>	<b><i>fPerpendicularLatitude</i></b>
<b>Descripción</b>	Determina la latitud en la que el ángulo de altura solar a las 12 del día es de 90 grados.
<b>Nombre</b>	<b><i>fHoursAngleOfSun</i></b>
<b>Descripción</b>	Se pasa como parámetro una hora y calcula el ángulo horario del Sol con respecto a la horizontal para la hora entrada (0°-90°).
<b>Nombre</b>	<b><i>fAzimuth</i></b>
<b>Descripción</b>	Se pasa una hora como parámetro y calcula el ángulo horario del Sol con respecto a la horizontal para la hora entrada, siempre medido desde el nacimiento del Sol (0°-180°).
<b>Nombre</b>	<b><i>fHeightAngleOfSun</i></b>
<b>Descripción</b>	Se pasa como parámetro una hora y calcula el ángulo de altura solar respecto a la horizontal para la hora entrada (0°-90°).
<b>Nombre</b>	<b><i>fOrientedHeightAngleOfSun</i></b>
<b>Descripción</b>	Se pasa como parámetro una hora y calcula el ángulo de altura solar respecto a la horizontal para la hora entrada, siempre medido desde el nacimiento del Sol (0°-180°).
<b>Nombre</b>	<b><i>pkConvertToCartesianCoordinates</i></b>
<b>Descripción</b>	Transforma de coordenadas esféricas a coordenadas cartesianas la posición del Sol, tomando como ángulos: el ángulo de altura solar y el ángulo horario de acuerdo a la hora que se pasa como parámetro.
<b>Nombre</b>	<b><i>PushSunInPosition</i></b>
<b>Descripción</b>	Se pasa como parámetro una hora y cambia la posición del Sol de acuerdo a esa hora.
<b>Nombre</b>	<b><i>PlayDay</i></b>

<b>Descripción</b>	Calcula la posición del Sol de acuerdo a la hora actual y además si la simulación automática de los efectos de precipitaciones y/o niebla está activada, pone en ejecución la estación que se corresponde con los datos de fecha y latitud del día.
<b>Nombre</b>	<b><i>UpdateDay</i></b>
<b>Descripción</b>	Traslada el Sol a la posición correspondiente con la hora actual en cada momento. Se encarga de que la geometría del Sol esté siempre de frente a la cámara. En dependencia de la hora del día y de las precipitaciones oculta o muestra la geometría del Sol en el momento que haga falta. Además actualiza la estación que se está ejecutando.
<b>Nombre</b>	<b><i>InitProbability</i></b>
<b>Descripción</b>	Calcula la probabilidad de ocurrencia de eventos de precipitaciones y/o nieblas, así como la cantidad y las características que tendrá cada uno de ellos. Además en caso de que haya eventos que se solapen, se hace un ajuste de la hora en que ocurrirán.
<b>Nombre</b>	<b><i>blsNight</i></b>
<b>Descripción</b>	Verifica si a la hora actual que se simula es o no de noche.
<b>Nombre</b>	<b><i>DisableSunLight</i></b>
<b>Descripción</b>	Deshabilita la luz del Sol.
<b>Nombre</b>	<b><i>EnableSunLight</i></b>
<b>Descripción</b>	Habilita la luz del Sol.
<b>Nombre</b>	<b><i>blsPrecipitation</i></b>
<b>Descripción</b>	Verifica si a la hora actual que se simula se está ejecutando o no un evento de precipitación.

Tabla 15 Descripción de la clase del diseño “CDay”.

<b>Nombre</b>	<b><i>CProbability</i></b>	
<b>Tipo de clase</b>	Entidad	
<b>Atributo</b>	<b>Tipo</b>	
<i>m_iProbabilityOfPrecip</i>	<i>int</i>	
<i>m_iEventQuantity</i>	<i>int</i>	
<i>m_afTimeX</i>	<i>float*</i>	
<i>m_afHourX</i>	<i>float*</i>	
<i>m_aiIntensityX</i>	<i>int*</i>	
<i>m_iProbabilityOfFog</i>	<i>int</i>	
<i>m_iEventQuantityOfFog</i>	<i>int</i>	
<i>m_afTimeFog</i>	<i>float*</i>	
<i>m_afHourOfFog</i>	<i>float*</i>	
<i>m_afDensityOfFog</i>	<i>float*</i>	
<b>Para cada responsabilidad</b>		
<b>Nombre</b>	<b><i>CProbability</i></b>	
<b>Descripción</b>	Constructor de la clase.	
<b>Nombre</b>	<b><i>Init</i></b>	



<b>Descripción</b>	Se pasa como parámetro la magnitud de una hora de simulación, la cual interviene en el tiempo de duración de los eventos de precipitación y/o niebla. Define la probabilidad de precipitación y/o niebla de un día y determina la cantidad de eventos de cada tipo que pueden ocurrir; además de cada uno de los eventos define el tiempo de duración, la intensidad y el tipo para el caso de las precipitaciones.
<b>Nombre</b>	<b><i>AdjustCalculations</i></b>
<b>Descripción</b>	Cambia la hora de ocurrencia de los eventos que comienzan cuando haya otro de su tipo ejecutándose, de forma tal que garantiza que sólo haya uno a la vez.
<b>Nombre</b>	<b><i>bHourOfPrecip</i></b>
<b>Descripción</b>	Se pasa una hora como parámetro y verifica si a esa hora comienza o no un evento de precipitación.
<b>Nombre</b>	<b><i>bHourOfFog</i></b>
<b>Descripción</b>	Se pasa una hora como parámetro y verifica si a esa hora comienza o no un evento de niebla.
<b>Nombre</b>	<b><i>OrderByHour</i></b>
<b>Descripción</b>	Ordena la hora de ocurrencia de los eventos de precipitación en orden ascendente.
<b>Nombre</b>	<b><i>OrderByHourOfFog</i></b>
<b>Descripción</b>	Ordena la hora de ocurrencia de los eventos de niebla en orden ascendente.
<b>Nombre</b>	<b><i>fRandomPrecipHour</i></b>
<b>Descripción</b>	Devuelve como resultado una hora cualquiera del día, encargándose de que haya una mayor probabilidad de que la hora esté en el intervalo de las 12 a las 18 horas.
<b>Nombre</b>	<b><i>fRandomFogHour</i></b>
<b>Descripción</b>	Devuelve como resultado una hora del día en el intervalo de las 22 horas a las 9 horas del día siguiente, encargándose de que haya una mayor probabilidad de que la hora esté en el intervalo de las 3 horas a las 9 horas.
<b>Nombre</b>	<b><i>iRandInt</i></b>
<b>Descripción</b>	Función auxiliar que devuelve un valor aleatorio entero, que pertenece a un rango definido por parámetros.
<b>Nombre</b>	<b><i>dRandFloat</i></b>
<b>Descripción</b>	Función auxiliar que devuelve un valor aleatorio entre 1 y 0.
<b>Nombre</b>	<b><i>dRandFloatInRange</i></b>
<b>Descripción</b>	Función auxiliar que devuelve un valor aleatorio tipo <b>float</b> , que pertenece a un rango definido por parámetros.

Tabla 16 Descripción de la clase del diseño "CProbability".

<b>Nombre</b>	<b>CSun</b>	
<b>Tipo de clase</b>	Entidad	
<b>Atributo</b>		<b>Tipo</b>
<i>m_pkPosition</i>		<i>CVector3*</i>
<i>m_fDistance</i>		<i>float</i>
<i>m_iValueOfVisibility</i>		<i>int</i>
<i>m_pkSunNode</i>		<i>CGroupNode*</i>
<i>m_pkSunLight</i>		<i>CLight*</i>
<i>m_pkSunGeometry</i>		<i>CGeometryNode*</i>
<i>m_pkLightState</i>		<i>CLightState*</i>
<i>m_bInIt</i>		<i>bool</i>
<i>m_fAuxTime</i>		<i>float</i>
<b>Para cada responsabilidad</b>		
<b>Nombre</b>	<b>CSun</b>	
<b>Descripción</b>	Constructor de la clase.	
<b>Nombre</b>	<b>HideSun</b>	
<b>Descripción</b>	Oculta la geometría del Sol.	
<b>Nombre</b>	<b>ShowSun</b>	
<b>Descripción</b>	Visualiza la geometría del Sol.	
<b>Nombre</b>	<b>ShowSunWaitSeconds</b>	
<b>Descripción</b>	Espera la cantidad de segundos pasada como parámetro para visualizar la geometría del Sol.	

Tabla 17 Descripción de la clase del diseño "CSun".

<b>Nombre</b>	<b>CSeason</b>	
<b>Tipo de clase</b>	Entidad	
<b>Atributo</b>		<b>Tipo</b>
<i>m_pkPrecipitation</i>		<i>CPrecipitation*</i>
<i>m_pkDay</i>		<i>CDay*</i>
<i>m_pkMyFog</i>		<i>CMyFog*</i>
<b>Para cada responsabilidad</b>		
<b>Nombre</b>	<b>CSeason</b>	
<b>Descripción</b>	Constructor de la clase.	
<b>Nombre</b>	<b>PlaySummer</b>	
<b>Descripción</b>	Ejecuta el comportamiento del clima en correspondencia con el verano.	
<b>Nombre</b>	<b>PlayAutumn</b>	
<b>Descripción</b>	Ejecuta el comportamiento del clima en correspondencia con el otoño.	
<b>Nombre</b>	<b>PlayWinter</b>	
<b>Descripción</b>	Ejecuta el comportamiento del clima en correspondencia con el invierno.	
<b>Nombre</b>	<b>PlaySpring</b>	
<b>Descripción</b>	Ejecuta el comportamiento del clima en correspondencia con la primavera.	
<b>Nombre</b>	<b>UpdateSeason</b>	
<b>Descripción</b>	Actualiza los eventos de precipitación y/o niebla.	

Tabla 18 Descripción de la clase del diseño "CSeason".

<b>Nombre</b>	<b>CPrecipitation</b>	
<b>Tipo de clase</b>	Entidad	
<b>Atributo</b>		<b>Tipo</b>
<i>m_apList_of_emiter</i>		<i>CGeometryNode **</i>
<i>m_iType</i>		<i>int</i>
<i>m_iIntensity</i>		<i>int</i>
<i>m_fTime</i>		<i>float</i>
<i>m_iEmitter_Count</i>		<i>int</i>
<i>m_bInIt</i>		<i>bool</i>
<i>m_fTimeC</i>		<i>float</i>
<i>m_fOscillator</i>		<i>float</i>
<i>m_pkRainCameraNode</i>		<i>CCameraNode*</i>
<i>kCameraLocation</i>		<i>CVector3</i>
<i>kCameraOrientation</i>		<i>CMatrix3</i>
<i>m_pkMainEmitter</i>		<i>CGroupNode *</i>
<b>Para cada responsabilidad</b>		
<b>Nombre</b>	<b>CPrecipitation</b>	
<b>Descripción</b>	Constructor de la clase	
<b>Nombre</b>	<b>PlayPrecipitation</b>	
<b>Descripción</b>	Según el tipo del evento de precipitación, lo ejecuta realizando una animación en la sección de textura correspondiente.	
<b>Nombre</b>	<b>UpdatePrecipitation</b>	
<b>Descripción</b>	Actualiza constantemente la animación de las texturas.	
<b>Nombre</b>	<b>StopPrecipitation</b>	
<b>Descripción</b>	Detiene el evento de precipitación que está en ejecución.	
<b>Nombre</b>	<b>InitPrecipitation</b>	
<b>Descripción</b>	Inicializa un evento de precipitación definiendo por parámetros el tipo de la precipitación, tiempo que durará, y la intensidad.	
<b>Nombre</b>	<b>AttachPrecipitationToCamera</b>	
<b>Descripción</b>	Adjunta el emisor de la precipitación a la cámara que se pasa por parámetros.	

Tabla 19 Descripción de la clase del diseño “CPrecipitation”.

<b>Nombre</b>	<b>CMyFog</b>	
<b>Tipo de clase</b>	Entidad	
<b>Atributo</b>		<b>Tipo</b>
<i>m_bInIt</i>		<i>bool</i>
<i>m_bEndFogState</i>		<i>bool</i>
<i>m_fPlayTime</i>		<i>float</i>
<i>m_fAuxTime</i>		<i>float</i>
<i>m_fFogDensity</i>		<i>float</i>
<b>Para cada responsabilidad</b>		
<b>Nombre</b>	<b>CMyFog</b>	

<b>Descripción</b>	Constructor de la clase.
<b>Nombre</b>	<b><i>InitFog</i></b>
<b>Descripción</b>	Inicializa el estado de niebla en la aplicación, atendiendo a los parámetros entrados. Se define la densidad y el tiempo de duración del evento de niebla.
<b>Nombre</b>	<b><i>PlayFog</i></b>
<b>Descripción</b>	Habilita el estado de niebla con los parámetros fijados en la inicialización.
<b>Nombre</b>	<b><i>StopFogSlowWay</i></b>
<b>Descripción</b>	Disminuye lentamente la densidad de la niebla hasta llevarla a un estado bien tenue, para entonces deshabilitarla y que no se observe un cambio brusco.
<b>Nombre</b>	<b><i>StopFog</i></b>
<b>Descripción</b>	Deshabilita el estado de niebla activo.
<b>Nombre</b>	<b><i>UpdateFog</i></b>
<b>Descripción</b>	Actualiza los cambios en las configuraciones del estado de niebla.

Tabla 20 Descripción de la clase del diseño "CMyFog".

### 4.3. Diagramas de secuencia

A continuación se presentan los diagramas de secuencia correspondientes a la expansión de cada caso de uso planteados en el capítulo anterior, a partir de los mismos se puede tener una idea más concreta de los mensajes que intercambian los objetos para realizar funciones específicas en el sistema.

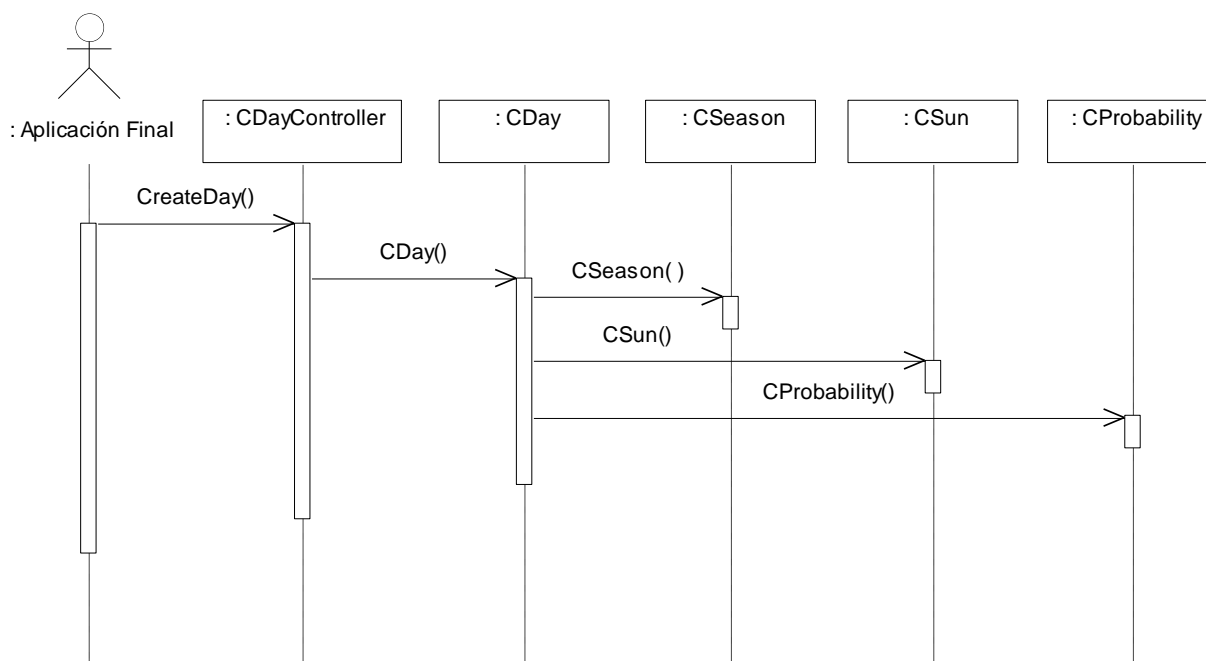


Fig. 17 Diagrama de secuencia CUS “Crear Día”.

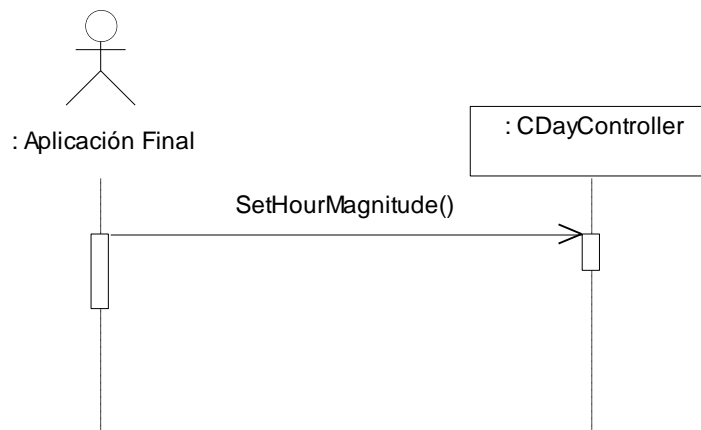


Fig. 18 Diagrama de secuencia CUS “Cambiar Magnitud Hora”.

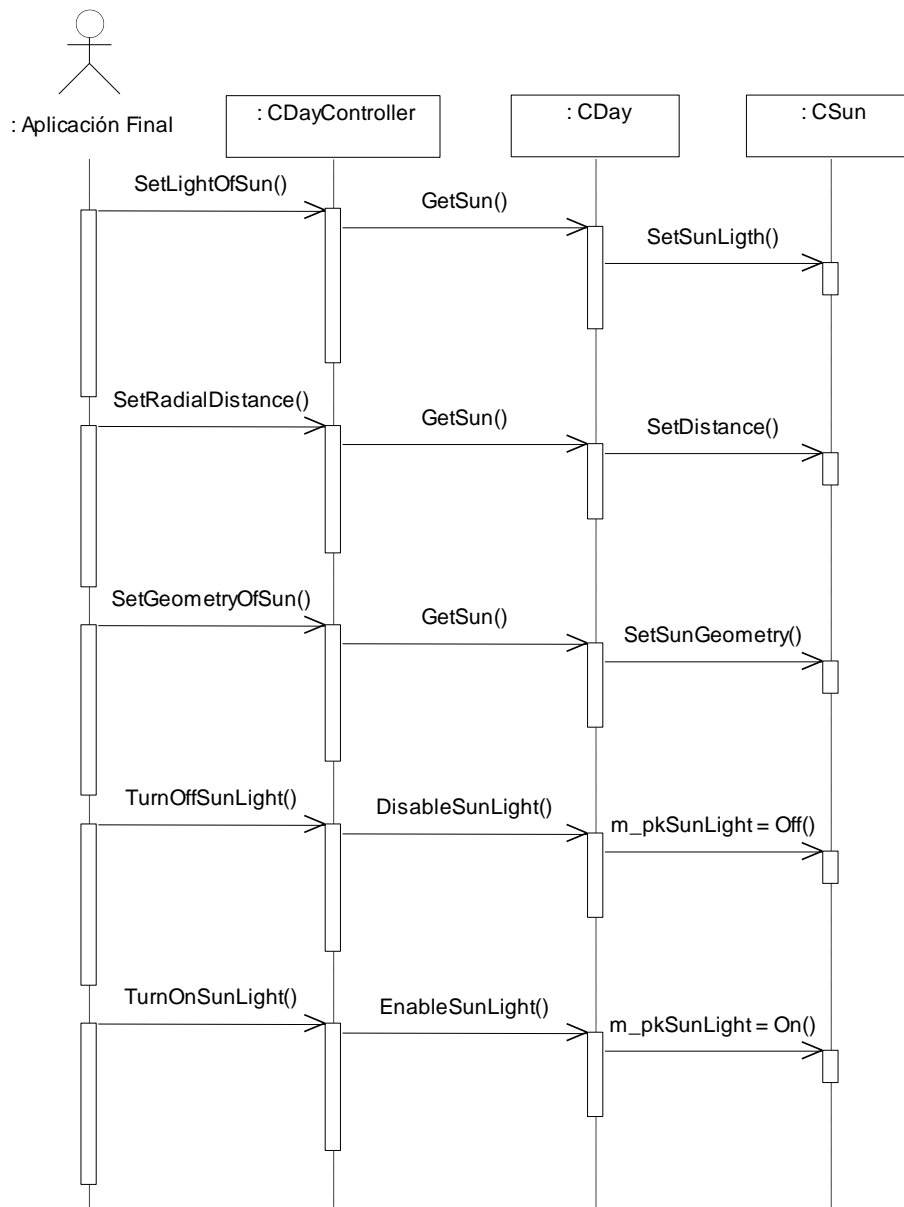


Fig. 19 Diagrama de secuencia CUS "Configurar Parámetros del Sol".

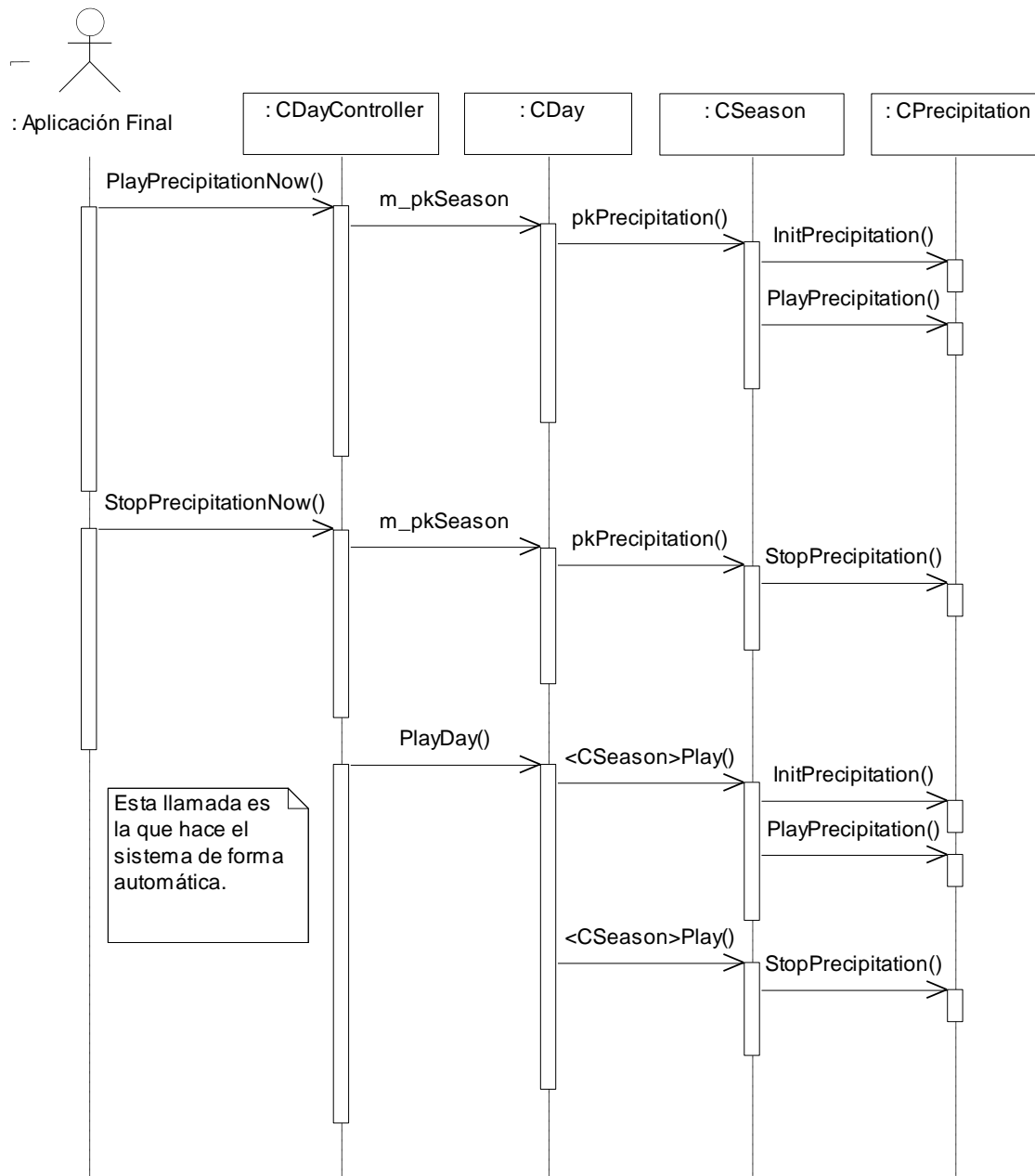


Fig. 20 Diagrama de secuencia CUS "Gestionar Precipitación".

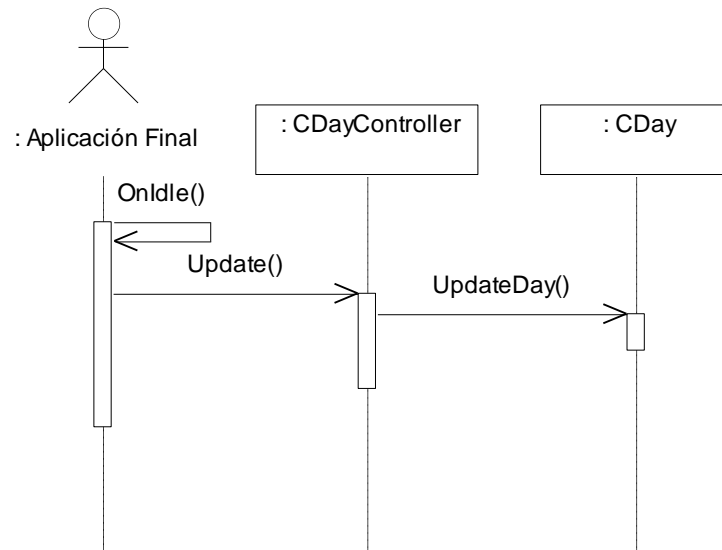


Fig. 21 Diagrama de secuencia CUS "Actualizar Día".

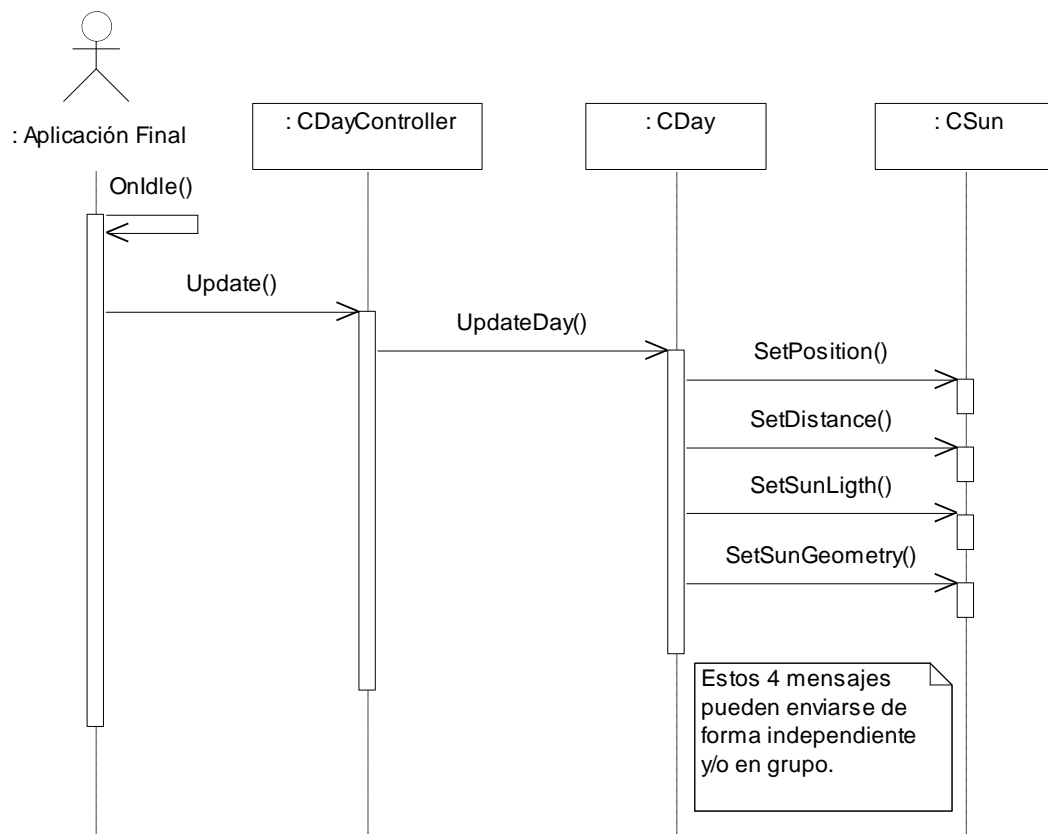


Fig. 22 Diagrama de secuencia CUS "Actualizar Sol".

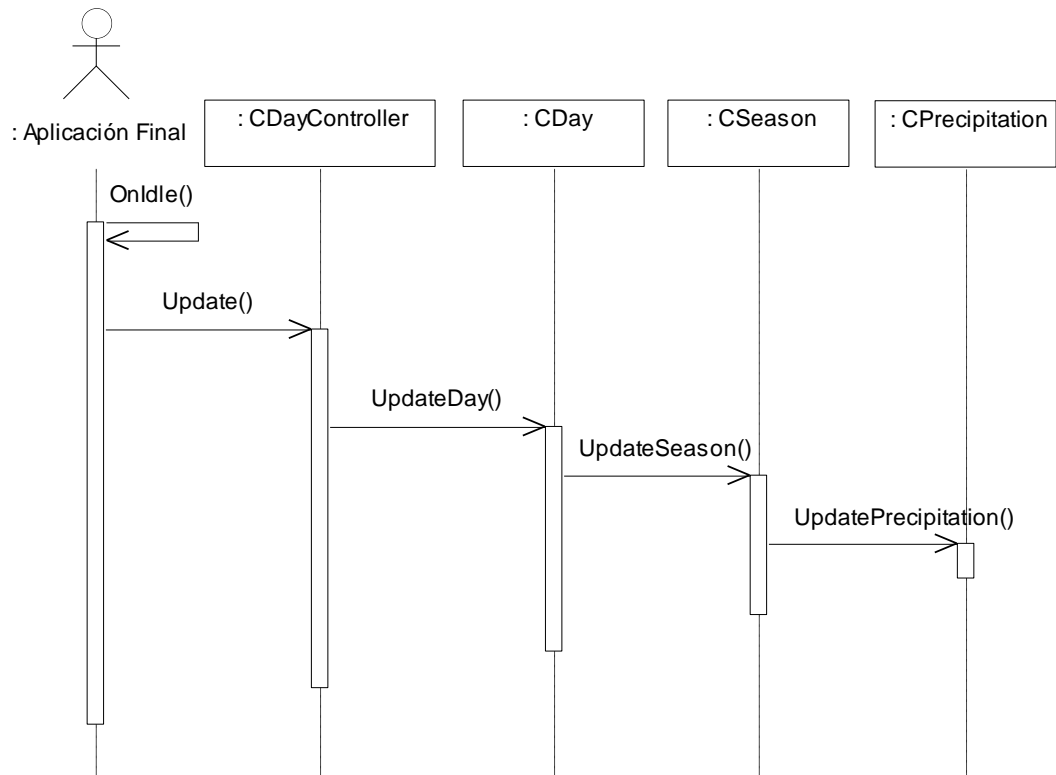


Fig. 23 Diagrama de secuencia CUS “Actualizar Precipitación”.

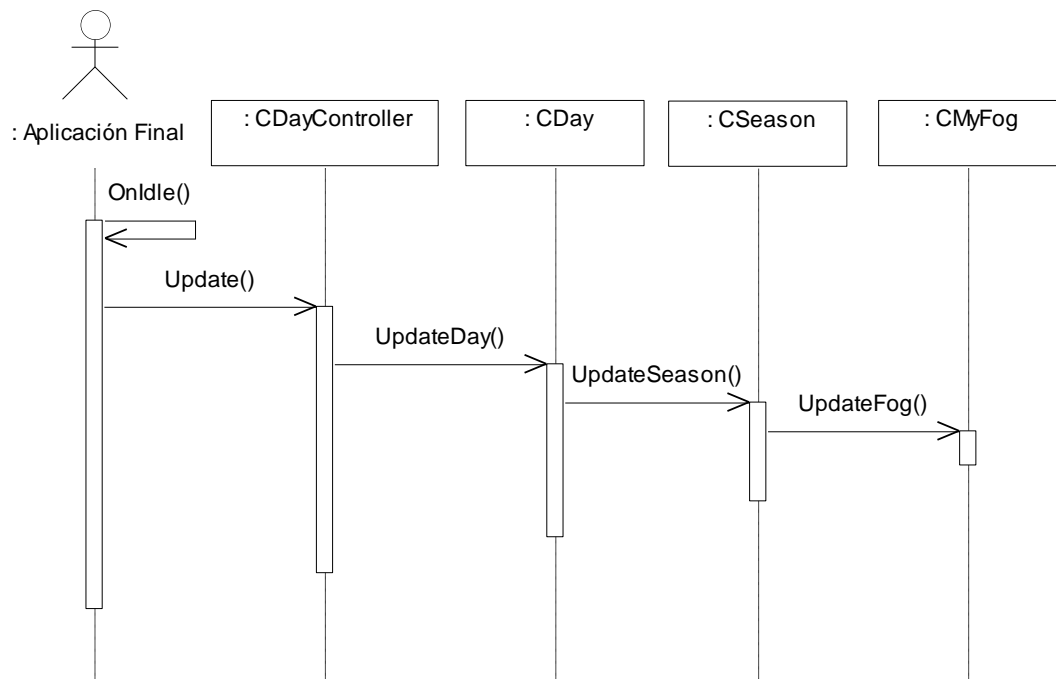


Fig. 24 Diagrama de secuencia CUS “Actualizar Niebla”.



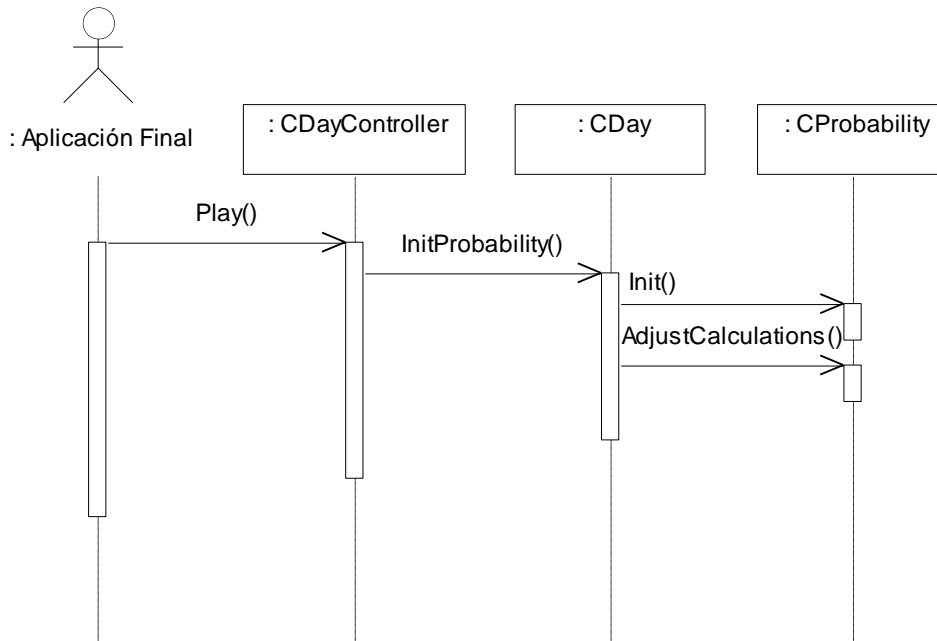


Fig. 25 Diagrama de secuencia CUS “Calcular Probabilidad de Precipitación o Niebla”.

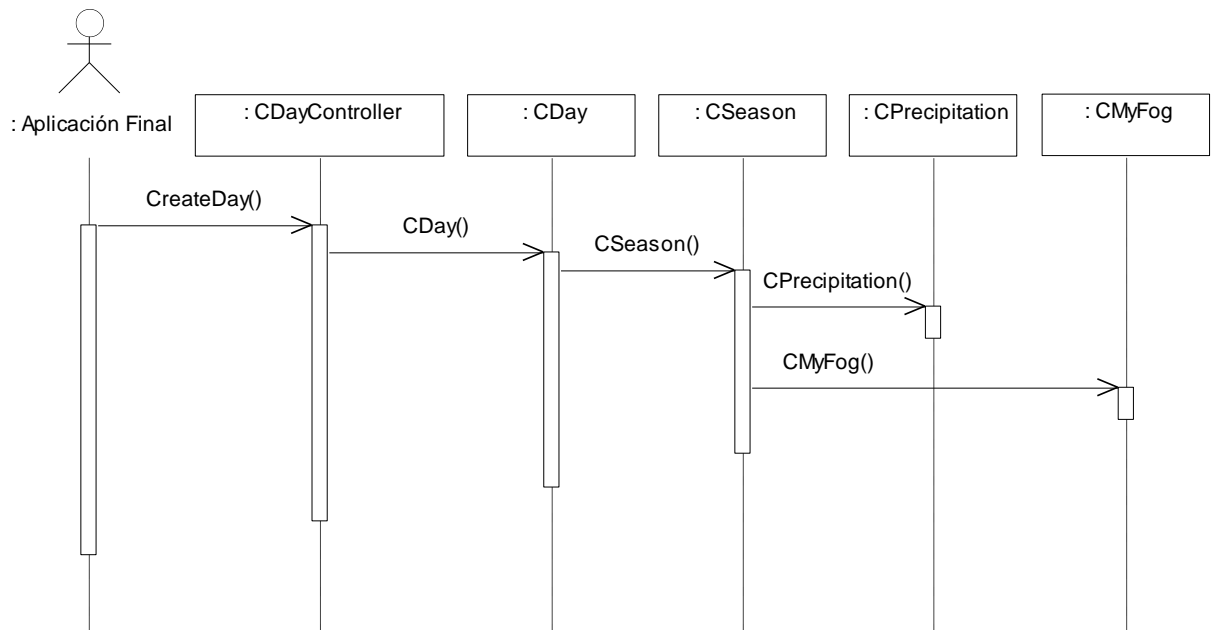


Fig. 26 Diagrama de secuencia CUS “Crear Estación”.

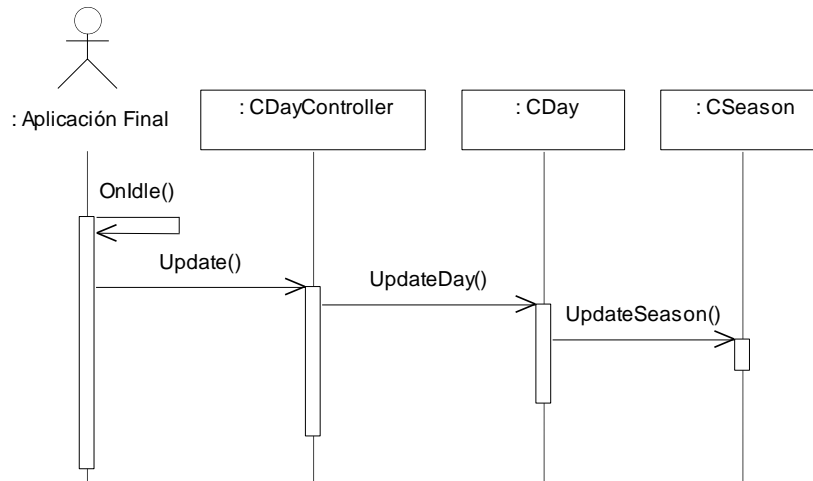


Fig. 27 Diagrama de secuencia CUS “Actualizar Estación”.

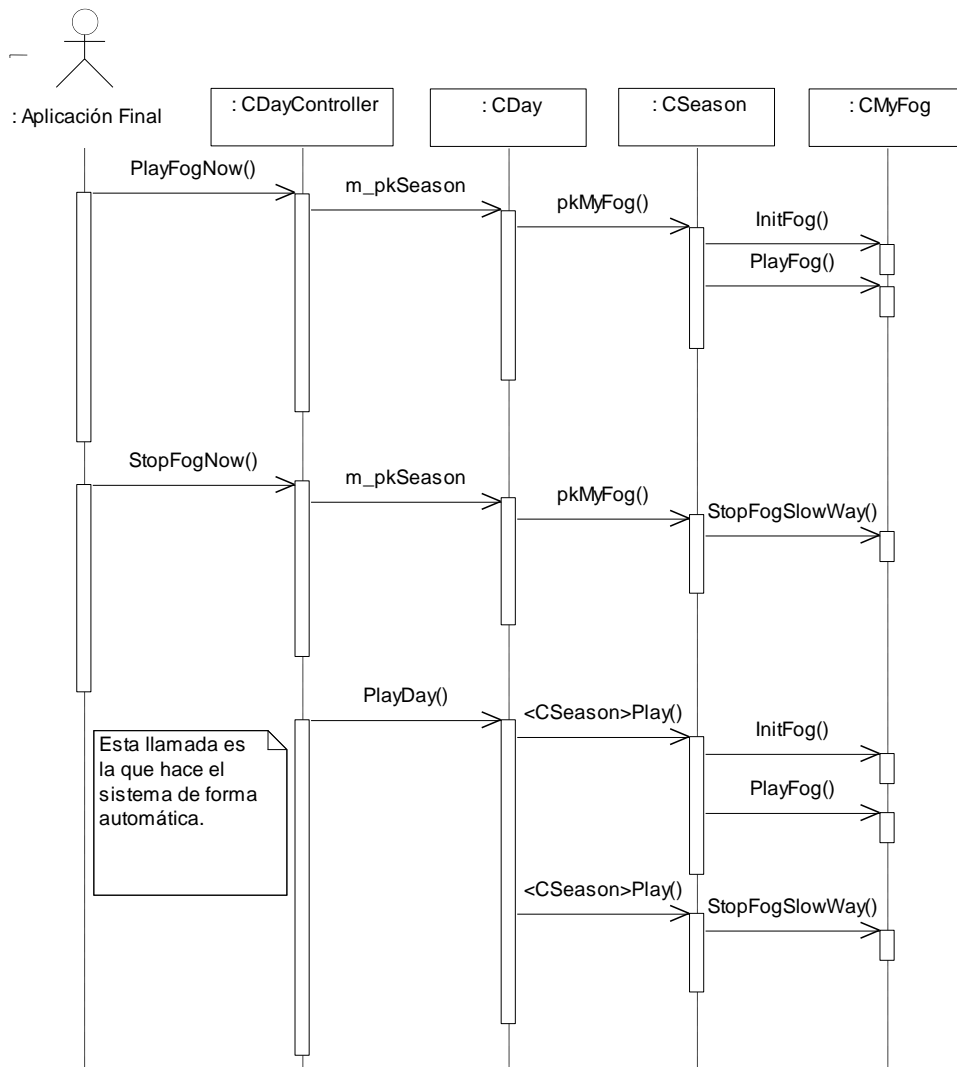


Fig. 28 Diagrama de secuencia CUS “Gestionar Niebla”.

### 4.4. Estándares de codificación

El código del módulo sigue algunos estándares propuestos por el grupo de desarrollo (respetando los estándares de codificación para C++). Está programado en inglés, debido a que las palabras son simples, no se acentúan y es un idioma muy difundido en el mundo informático.

El conocimiento de los estándares que vienen a continuación permitirá un mayor entendimiento del código fuente.

#### Nombre de los ficheros:

Los nombres de los ficheros .h y .cpp utilizan las iniciales del nombre de la herramienta (STK).

Ej: STKNameOfUnits.cpp

#### Tipos de datos:

Los tipos se nombrarán siguiendo el siguiente patrón:

- **Constantes:** Las constantes se nombran con mayúsculas, utilizándose “\_” para separar las palabras.

Ej: `const int MY_CONST_ZERO = 0;`

- **Enumerados:** Para los enumerados se utiliza el indicador “E” en el nombre del tipo, y en las constantes se utilizan las iniciales del nombre del enumerado. Las constantes van en mayúsculas.

Ej1: `enum EMyEnum`

```
{  
    ME_VALUE,  
    ME_OTHER_VALUE  
};
```

Ej2: `enum ENodeType`

```
{  
    NT_GEOMETRYNODE,  
    NT_GROUPNODE...  
};
```

- **Estructuras:** Se utiliza el indicador “S” para indicar que es una estructura. Las variables miembros de la estructura se nombrarán igual que en las clases, leer más adelante.

Ej: struct SMyStruct {...};

### Clases:

Se utiliza el indicador “C” para indicar que es una clase. Ver más adelante la nomenclatura de las variables miembros.

Ej: class CClassName;

### Interfaces:

Se utiliza el indicador “I” para indicar que es una interfaz.

Ej: IMyInterfaceName

### Listas e iteradores de la std:

Para los tipos de datos utilizados de la librería estándar de C++ (vector, map, multimap, etc.), se utiliza el indicador “T”, con los sufijos List, Map y MultiMap según la estructura, así como el sufijo lter para los iteradores. Además el nombre lleva el tipo de dato a almacenar en la estructura en cuestión:

Ej: vector<CNode> TNodeList;

TNodeList::iterator TNodeListlter;

Ej: map<> TNameMap;

TNameMap::iterator TnameMaplter;

Ej: multimap<> TNameMultiMap;

TNameMultiMap::iterator TNameMultiMaplter;

### Declaración de variables:

Los nombres de las variables comienzan con un identificador del tipo de dato al que correspondan, como se muestra a continuación. En el caso de que sean variables miembros de una clase, se le antepone el identificador “m\_” (en minúscula), si son globales se les antepone el identificador “g\_”, y en caso de ser argumentos de algún método, se les antepone el prefijo “arg\_”.

– **Tipos simples:**

```
Ej: bool bVarName;
    int iName;
    unsigned int uiName;
    float fName;
    char cName;
    char* acName; // arreglo de caracteres
    char* pcName; // puntero a un char
    char** aacName; // bidimensional
    char** apcName; // arreglo de punteros
    bool m_bMemberVarName; // variable miembro de una estructura o clase
    char g_cGlobalVarName; // variable global
    short sName;
    void* pvName;
```

– **Instancias de tipos creados:**

```
Ej: EMyEnumerated eName;
    SMyStructure kName;
    CClassName kObjectName; // objeto de una clase
    CClassName* pkName;     // puntero a objeto
    CClassName* akName;     // arreglo de objetos
    CClassName* m_akName; // variable miembro de claseIMyInterface*
    pIName; // puntero interfaces
```

### Métodos:

En el caso de los métodos, se les antepone el identificador del tipo de dato de devolución, y en caso de no tenerlo (void), no se les antepone nada. Los constructores y destructores, como lo exigen los compiladores, llevan el nombre de la clase.

– **Constructor y destructor:**

```
Ej: CClassName (bool arg_bVarName, float& arg_fVarName);
    ~CClassName ();
```

– **Funciones:**

```
Ej: bool bFunction1 (...);
    int* piFunction2 (...);
    CClassName* pkFunction3 (...);
```

– **Procedimientos:**

```
Ej: void Procedure4 (...);
```

### Métodos de acceso a miembros:

Los métodos de acceso a los miembros de las clases no se nombrarán “Gets” ni “Sets”, sino como los demás métodos, pero con el nombre de la variable a la que se accede y sin el prefijo “m\_”:

Ej: Para la siguiente variable, los métodos de acceso serían:

```
intm_iMyVar; //variable  
int iMyVar(); //”Get”  
void MyVar(int arg_iMyVar) //”Set”  
int& iMyVar(); //”Get” y ”Set”
```

### 4.5. Diagrama de despliegue

El diagrama de despliegue para este sistema es muy simple, incluye sólo la representación de una PC debido a que todos los componentes para la ejecución del módulo se encontrarán en la misma máquina, por esta razón se decidió que no es necesario representar dicho diagrama.

### 4.6. Diagrama de componentes

Para lograr un mejor entendimiento de la implementación del módulo se muestran los componentes .h y .cpp que encapsulan el código fuente de éste, así como las relaciones de dependencia entre ellos a través del diagrama siguiente.

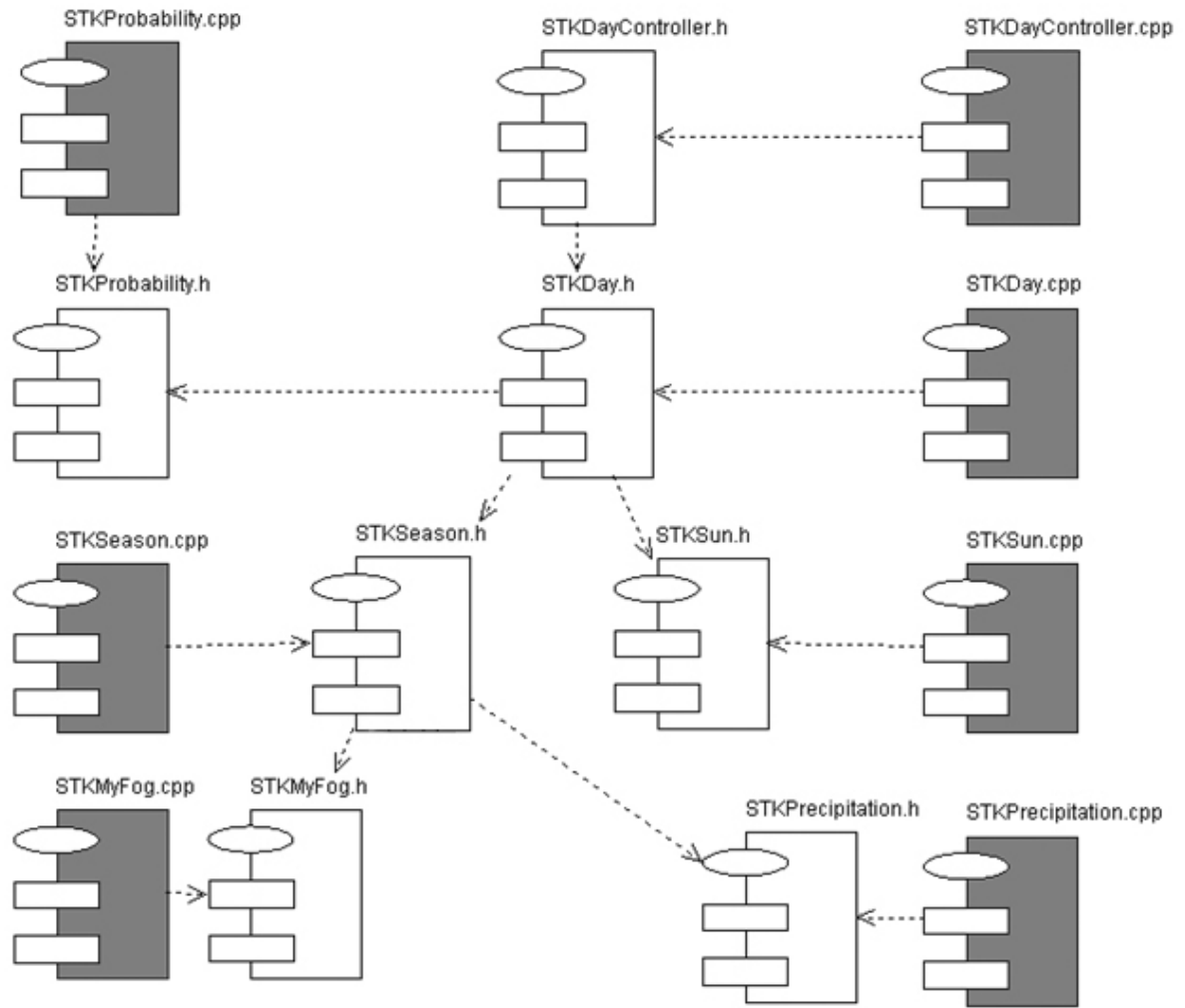


Fig. 29 Diagrama de Componentes.

# Conclusiones

---

Se desarrolló un módulo de clases capaz de simular las estaciones y los horarios dentro de los Entornos de Realidad Virtual que se generen con la herramienta SceneToolkit, y de esta forma se dio cumplimiento a los objetivos propuestos para el trabajo. Los logros más significativos que se pueden resaltar de este módulo de listan a continuación.

Las aplicaciones que utilicen el módulo podrán simular efectos climatológicos tales como lluvia, nieve, y niebla sin que el programador tenga que realizar un gran esfuerzo mental, ni efectuar exhaustivos cálculos probabilísticos; porque la aplicación cuenta con numerosas funciones para personalizar la simulación y adaptarse a disímiles situaciones climáticas.

Independientemente del hardware utilizado para la ejecución de una aplicación que haga uso del presente módulo, el sistema siempre ubicará el Sol en la posición correcta en dependencia de los parámetros establecidos (día, mes, latitud, hora, etc.) y realizará el cálculo probabilístico propio de la estación que se esté simulando. Aunque se recomienda respetar los requerimientos de hardware especificados en el documento para alcanzar el máximo de rendimiento y fluidez.

Fue diseñado de una manera flexible, para posibilitar futuras expansiones, en vísperas de incorporar nuevos efectos climatológicos, y con ello, nuevas funcionalidades que harán cada vez más fácil la simulación de las estaciones y los horarios.



# Recomendaciones

---

Se recomiendan los siguientes aspectos:

- Agregar otros efectos ambientales asociados a las estaciones.
- Determinar la distancia del Sol al observador a partir de datos perimetrales del mundo.
- Adicionar sonido a los efectos ambientales para aumentar el nivel de realismo.
- Incluir acumulación de la precipitación.

# Referencias Bibliográficas

---

- [1]. **Casero, Jesús A. Manzaneque.** El planeta Tierra.  
[[http://almez.pntic.mec.es/~jmac0005/ESO\\_Geo/TIERRA/Html/Movimientos\\_b.htm](http://almez.pntic.mec.es/~jmac0005/ESO_Geo/TIERRA/Html/Movimientos_b.htm)].
- [2]. —. El planeta Tierra.  
[[http://almez.pntic.mec.es/~jmac0005/ESO\\_Geo/TIERRA/Html/Movimientos\\_c.htm](http://almez.pntic.mec.es/~jmac0005/ESO_Geo/TIERRA/Html/Movimientos_c.htm)].
- [3]. Realidad Virtual. [<http://paguzmancortes.googlepages.com/realidadvirtual>].
- [4]. **Galván, Isabel Fonseca.** Climas y Paisajes.  
[<http://thales.cica.es/rd/Recursos/rd99/ed99-0151-01/capitulos/cap3.htm>].
- [5]. **Roing, Vicente Luis Rosell.** Departamento de Sistemas Informáticos y Computación. Enero de 2006. [<http://www.dsic.upv.es/docs/bib-dig/tesis/etd-11092005-145237/TesisdefinitivaRosellRoig.pdf>].
6. **McReynolds, Tom y Blythe, David.** *Advanced Graphics Programming Techniques Using OpenGL*. 1998. Chapter 8. Lighting. 8.4 Global Illumination.
- [7]. Departamento de Informática, Universidad de Valencia. 2008.  
[[http://informatica.uv.es/iiguia/AIG/web\\_teoría/tema2.pdf](http://informatica.uv.es/iiguia/AIG/web_teoría/tema2.pdf)].
- [8]. **Foley y Dam, Van.** Dynamic Graphics Project.  
[[http://www.dgp.toronto.edu/~ah/csc418/fall\\_2001/notes/illum\\_local.html](http://www.dgp.toronto.edu/~ah/csc418/fall_2001/notes/illum_local.html)]. p. 721-741, 760-766.
- [9]. CSAIL. *Local Illumination Models*. 17 de Septiembre de 2002.  
[<http://groups.csail.mit.edu/graphics/classes/6.837/F02/lectures/17sep2002.pdf>].
- [10]. *WorldToolkit Release 9, Reference Manual*. [www.sense8.com] USA : s.n., 1999. Lights Chapter.
- [11]. **Marcos, Mercedes Sánchez y García, Almudena Sardón.** 2007.  
[<http://gsii.usal.es/~corchado/igrafica/descargas/temas/Tema08.ppt>].
- [12]. **Birn, J.** *Iluminación Exterior*.  
[<http://xuanprada.net/material/articulos/ilumExterior.pdf>] España : s.n., 2007.
- [13]. **McReynolds, Tom y Blythe, David.** *Advanced Graphics Programming*

*Techniques Using OpenGL*. 1998. Chapter 11 Natural Phenomena 11.9.6 Particle Systems in a Scene.

[14]. —. *Advanced Graphics Programming Techniques Using OpenGL*. 1998. Chapter 11. Natural Phenomena 11.9 Particle Systems.

[15]. **Wang, Niniane y Wade, Bretton**. *Rendering Falling Rain and Snow*. [<http://niniane.org/rainsnow/rainsnow-sketch.pdf>] 2003.

[16]. Departamento de Ciencias Ambientales, Universidad Rey Juan Carlos de Madrid. *Meteorología y Climatología*.

[[http://www.escet.urjc.es/fisica/Docencia/CCAA/meteo/practicas/practica\\_radiacion.pdf](http://www.escet.urjc.es/fisica/Docencia/CCAA/meteo/practicas/practica_radiacion.pdf)] 2006-2007. Capítulo 3 Cálculo de la insolación diaria.

[17]. VJuegos: Comunidad Iberoamericana de Desarrolladores de juegos. 2004. [<http://www.vjuegos.org>].

[18]. **Astle, Dave y Hawking, Kevin**. *OpenGL Game Programming*. Capítulo 19. Physics Modeling with OpenGL. Modeling the Real World. USA : s.n., 2001.

[19]. **McGuire, Morgan**. *Manual de G3D versión 6.10* . Julio 2006.

[20]. G3D-cpp. 2007. [[http://g3d-cpp.sourceforge.net/manual/classG3D\\_1\\_1Sky.html](http://g3d-cpp.sourceforge.net/manual/classG3D_1_1Sky.html)].

[21]. *Ogre 3D*. [<http://alts.homelinux.net/libreapp.php?id=574>]. 2007

[22]. *Particle System Attributes*. [[http://www.ogre3d.org/docs/manual/manual\\_32.html](http://www.ogre3d.org/docs/manual/manual_32.html)]. 2007.

[23]. **Antonio Gros, Ceuta**. [<http://www.ugr.es/~agros/ugr/fisica34.htm>] España: s.n.

[24] *Origen de la niebla*. [<http://www.conciencia-nimal.cl/paginas/temas/temas.php>]. Junio de 2006.

# Bibliografía Consultada

---

- **McReynolds, Tom y Blythe, David.** *Advanced Graphics Programming Techniques Using OpenGL.* 1998.
- **McGuire, Morgan.** *G3D Manual.* V6.10. Julio 2006.
- **Eberly, David H.** *3D Game Engine Architecture.* 2005.
- Documentación para usuarios de la herramienta *SceneToolKit V2.3.* Noviembre 2007.
- *Geografía Física* (Tercera Edición). 1976.
- **Wang, Niniane y Wade, Bretton.** *Rendering Falling Rain and Snow.* 2003.
- **Felicísimo, Angel M.** Simulación de procesos: modelización climática.  
[[http://www.etsimo.uniovi.es/~feli/CursoMDT/Tema\\_6.pdf](http://www.etsimo.uniovi.es/~feli/CursoMDT/Tema_6.pdf)]

# Glosario de Términos

---

**C++:** Lenguaje que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos (POO). C++ está considerado por muchos como uno de los lenguajes más potentes, debido a que permite trabajar tanto a alto como a bajo nivel.

**Cortina:** Conjunto de geometrías que se distribuyen paralelamente en frente del observador. Se utiliza con el objetivo de disminuir la cantidad de geometría a renderizar en una escena.

**Engine:** Programa orientado a la creación de otros *softwares*.

**Entorno de Realidad Virtual:** Sistema informático interactivo que ofrece una percepción sensorial al usuario de un mundo tridimensional sintético que suplanta al real.

**Evento de precipitación/niebla:** Manifestación de precipitaciones/niebla que contiene características propias dentro de la simulación.

**Frame:** Cada uno de las imágenes que componen una animación.

**Framerate:** Número de *frames* por segundo.

**Hardware:** Es la parte física de un computador y más ampliamente de cualquier dispositivo electrónico.

**Herramienta:** Colección de herramientas integradas que permiten automatizar un conjunto de tareas de algunas de las fases del ciclo de vida del sistema informático: Planificación estratégica, Análisis, Diseño, Generación de programas.

**Hora de simulación:** Abstracción de una hora real. Se mide en segundos y puede durar entre 30 y 3600 segundos (una hora real).

**Hora local:** Es una medida de la posición del Sol relativa a una localidad, de tal forma que las 12 horas locales se corresponde con la posición más alta del Sol sobre el horizonte.

**Horas de luz:** Cantidad de horas que transcurren entre la salida y puesta del Sol.

**Magnitud de una hora:** Tiempo en segundos que dura una hora de simulación.

**Mundo virtual:** Puede considerarse como el paradigma que actualmente rige la construcción de modelos basados en Realidad Virtual. Proponiendo un cambio fundamental en la naturaleza de la denominada interfaz usuario (que rige la orientación de la interacción hombre-máquina), desplazándola hacia un diseño centrado en lo humano, según el cual el espacio alrededor del usuario se constituye en el ambiente de computación y una entera gama de percepciones sensoriales se conjuga en torno a la nueva interfaz. Todo esto se traduce en un esfuerzo por hacer que la tecnología de computación se haga más amistosa y accesible al usuario enfocándose hacia un planteamiento básico: si algo puede ser representado sensorialmente, es posible incorporarlo al medio computarizado.

**Realidad Virtual:** Término futurista el cual pretende describir la interacción de los seres humanos en mundos virtuales o simulados.

**Render:** Proceso mediante el cual el ordenador crea una imagen partiendo de la descripción de las características de los objetos a visualizar.

**Rendering:** Crear en forma automática una imagen de acuerdo al modelo tridimensional que existe en el ordenador.

**Renderizar:** Hacer render.

**Sección de textura:** Parte de una textura diseñada para simular un tipo específico de precipitación.

**Simulación:** Programa computacional basado en cálculos y modelos estadísticos, usados para representar un escenario determinado.

**Simulación continua:** Simulación del transcurso del tiempo que garantiza la sucesión de los días y las noches.

# Anexos

---

## Anexo 1

El módulo utiliza planos como **geometrías** base para la representación de las precipitaciones y el Sol. Dichas geometrías deben exportarse en formato 3DX y con las siguientes características:

1. Estar ubicadas en el origen de coordenadas del 3DS Max Studio ( $x=0$ ,  $y=0$ ,  $z=0$ ) y ubicados de forma frontal al visor "Left" de dicho programa (orientado hacia las X negativas).
2. Poseer un mapeo básico en sus coordenadas de textura, para evitar deformaciones no deseadas en la textura.
3. Las dimensiones son acorde a las necesidades del usuario final, no obstante se recomiendan las siguientes:
  - Plano que representa el **Sol**: *80 cm x 80 cm*
  - Plano que representa las **precipitaciones**: *110 cm x 110 cm*

Para las **texturas** que se aplicarán a estos objetos hay que tener en cuenta que deben exportarse en formato TGA y para el caso de la precipitación tener en cuenta que debe ser una textura lo suficientemente alta como para dividirla en 8 secciones que representan los estados básicos de precipitación. (Se recomienda: *Ancho 128 x Alto 6096*).

8	1. El inicio de la lluvia.
7	2. La llovizna.
6	3. La lluvia en un estado más fuerte.
5	4. Fin del evento de lluvia.
4	5. El inicio de la nieve.
3	6. La nieve en su manifestación débil.
2	7. La nieve más fuerte.
1	8. Fin del evento de nieve.

Fig. 30 Textura de precipitaciones.

## Anexo 2

La orientación geográfica que utilizara el módulo de clases está basada en la orientación que establece el ayudante “Compass” del 3DS Max Studio. De esta forma el Este siempre quedará ubicado para la parte positiva de las X, y el Norte para la parte positiva de las Y. Teniendo en cuenta estas características, se garantiza una correcta ubicación a la hora de simular la trayectoria del Sol, como se muestra en la siguiente figura.

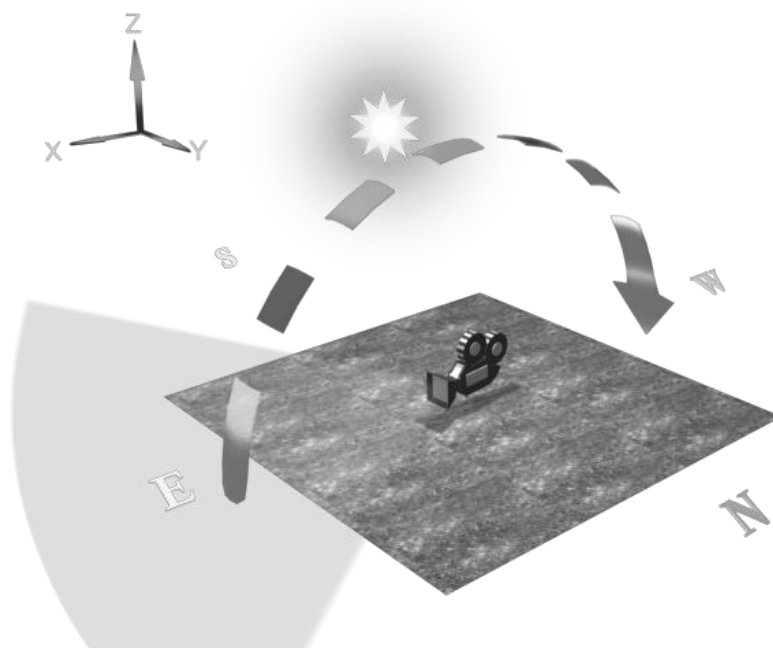


Fig. 31 Orientación geográfica.



## Índice de Tablas

Tabla 1 Descripción del actor del sistema. ....	39
Tabla 2 Descripción del CU Crear día. ....	42
Tabla 3 Descripción del CU Cambiar Magnitud Hora. ....	43
Tabla 4 Descripción del CU Configurar Parámetros del Sol. ....	43
Tabla 5 Descripción del CU Gestionar Precipitación. ....	44
Tabla 6 Descripción del CU Actualizar Día. ....	45
Tabla 7 Descripción del CU Actualizar Sol. ....	45
Tabla 8 Descripción del CU Actualizar Precipitación. ....	46
Tabla 9 Descripción del CU Actualizar Niebla. ....	46
Tabla 10 Descripción del CU Calcular Probabilidad de Precipitación o Niebla. ....	47
Tabla 11 Descripción del CU Crear Estación. ....	47
Tabla 12 Descripción del CU Actualizar Estación. ....	48
Tabla 13 Descripción del CU Gestionar Niebla. ....	49
Tabla 14 Descripción de la clase del diseño "CDayController" ....	54
Tabla 15 Descripción de la clase del diseño "CDay". ....	56
Tabla 16 Descripción de la clase del diseño "CProbability". ....	57
Tabla 17 Descripción de la clase del diseño "CSun". ....	58
Tabla 18 Descripción de la clase del diseño "CSeason". ....	58
Tabla 19 Descripción de la clase del diseño "CPrecipitation". ....	59
Tabla 20 Descripción de la clase del diseño "CMyFog" ....	60

## Índice de Figuras

Fig. 1 Puntos Cardinales .....	5
Fig. 2 Coordenadas geográficas: Latitud y Longitud. ....	6
Fig. 3 Husos horarios.....	7
Fig. 4 Solsticio de Verano. ....	8
Fig. 5 Equinoccio de Otoño.....	8
Fig. 6 Solsticio de Invierno.....	9
Fig. 7 Equinoccio de Primavera.....	9
Fig. 8 Modelo de iluminación global. ....	12
Fig. 9 Modelo de iluminación local. ....	13
Fig. 10 Tabla de partículas. ....	16
Fig. 11 Representación del cono doble y de una textura de lluvia. ....	17
Fig. 12 Coordenadas de posicionamiento del Sol respecto de un punto en la Tierra en el hemisferio norte. ....	18
Fig. 13 Referencias celestes con las coordenadas del Sol respecto de un observador en la Tierra. ....	19
Fig. 14 Modelo del Dominio. ....	34
Fig. 15 Diagrama de casos de uso del sistema.....	40
Fig. 16 Diagrama de clases de diseño. ....	52
Fig. 17 Diagrama de secuencia CUS “Crear Día”.....	61
Fig. 18 Diagrama de secuencia CUS “Cambiar Magnitud Hora”.....	61
Fig. 19 Diagrama de secuencia CUS “Configurar Parámetros del Sol”.....	62
Fig. 20 Diagrama de secuencia CUS “Gestionar Precipitación”.....	62
Fig. 21 Diagrama de secuencia CUS “Actualizar Día”.....	63
Fig. 22 Diagrama de secuencia CUS “Actualizar Sol”.....	63
Fig. 23 Diagrama de secuencia CUS “Actualizar Precipitación”.....	64
Fig. 24 Diagrama de secuencia CUS “Actualizar Niebla”.....	64

Fig. 25 Diagrama de secuencia CUS “Calcular Probabilidad de Precipitación o Niebla” .....65

Fig. 26 Diagrama de secuencia CUS “Crear Estación” .....65

Fig. 27 Diagrama de secuencia CUS “Actualizar Estación” .....66

Fig. 28 Diagrama de secuencia CUS “Gestionar Niebla” .....66

Fig. 29 Diagrama de Componentes .....71

Fig. 30 Textura de precipitaciones .....79

Fig. 31 Orientación geográfica .....80